

AD-A102 619

L N K CORP SILVER SPRING MD

F/6 9/2

STUDY OF DIGITAL MATCHING OF DISSIMILAR IMAGES. (U)

NOV 80 B A LAMBIRD, D LAVINE, G C STOCKMAN

DAAK70-79-C-0234

UNCLASSIFIED

ETL-0248

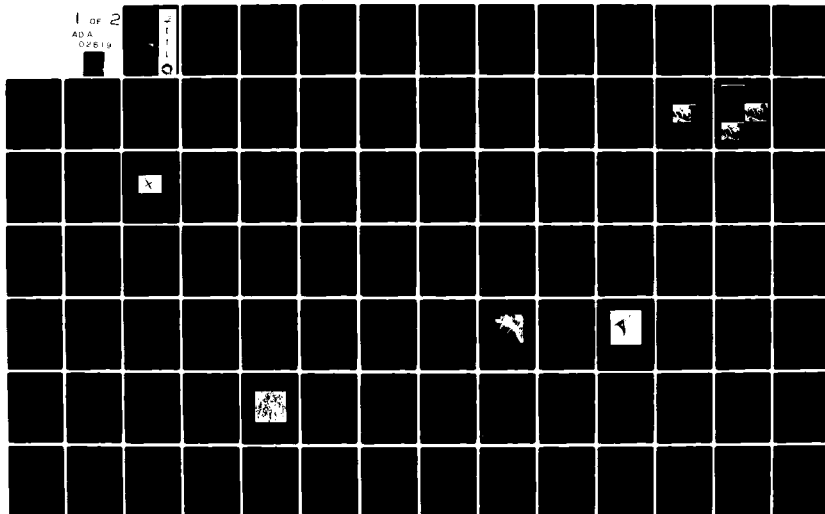
NL

1 OF 2

AD A

02619

0



ETL-0248

**LEVEL**

(12)

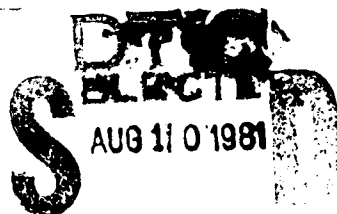
AD A102619

Study of digital matching  
of dissimilar images

Barbara A. Lambird  
David Lavine  
George C. Stockman  
Kenneth C. Hayes  
Laveen N. Kanal

L.N.K. Corporation  
302 Notley Court  
Silver Spring, MD 20904

NOVEMBER 1980

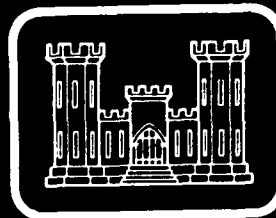


DTIC FILE COPY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Prepared for  
U.S. ARMY CORPS OF ENGINEERS  
ENGINEER TOPOGRAPHIC LABORATORIES  
FORT BELVOIR, VIRGINIA 22060

81 8 10 038



E

T

L



Destroy this report when no longer needed.  
Do not return it to the originator.

---

The findings in this report are not to be construed as an official  
Department of the Army position unless so designated by other  
authorized documents.

---

The citation in this report of trade names of commercially available  
products does not constitute official endorsement or approval of the  
use of such products.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETLH0248	2. GOVT ACCESSION NO. AD-A102	3. RECIPIENT'S CATALOG NUMBER 619
4. TITLE (and Subtitle) STUDY OF DIGITAL MATCHING OF DISSIMILAR IMAGES		5. TYPE OF REPORT & PERIOD COVERED Contract Report
7. AUTHOR(s) Barbara A. Lambird, David Lavine, George C. Stockman, Kenneth C. Hayes, and Laveen N. Kanal		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS L.N.K. Corporation 302 Notley Court Silver Spring, MD 20904		8. CONTRACT OR GRANT NUMBER(s) DAAK70-79-C-0234
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineering Topographic Laboratories Fort Belvoir, Virginia 22060		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060		12. REPORT DATE 11 November 1980
		13. NUMBER OF PAGES 117
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) registration, image matching, automated cartography, feature extraction, correlation, pattern recognition, image analysis, dissimilar images.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) In this report, the registration of digital images from similar and dissimilar sensors is studied. The matching problem is divided into four subproblems: 1. detecting and extracting appropriate features; 2. determining an approximate global registration; 3. determining disparity for a subset of points in the image; and (4) determining the global nonlinear transformation for the entire image. Three classes of features and their detectors are considered; edge features, point features, and region features. Several registration methods are		

DD FORM 1473, JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

described, including a number of correlation procedures, the L.N.K. registration technique and its 3-d extension, and some region matching procedures. The registration techniques are briefly compared and contrasted. Their ability to handle the full rotation, scale and translation transformations, and their time and space requirements are considered. It is concluded that only two registration techniques are worthy of further investigation: the L.N.K. registration technique and a combined hierarchical-sequential correlation method using edge images.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

### Preface

The work presented in this report was performed by L.N.K. Corporation scientists Kenneth C. Hayes, Laveen N. Kanal, Barbara A. Lambird, David Lavine, and George C. Stockman under contract DAAK70-79-C-0234 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia 22060. Mr. A. T. Blackburn, Computer Sciences Division, USAETL served as contact technical monitor. The authors are indebted to Mr. Blackburn and Mr. Michael Crombie, Computer Science Division, USAETL for helpful interaction and technical reviews during the performance of this work.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A	

# Study of Digital Matching of Dissimilar Images

## Table of Contents

Preface . . . . .	i
1. Introduction and Summary . . . . .	1
2. Rectification . . . . .	5
3. Feature Selection . . . . .	7
3.1 Feature Detection . . . . .	7
3.2 Edge Detection . . . . .	8
3.2.1 First Derivative Methods . . . . .	10
3.2.2 Marr Edge Detector . . . . .	12
3.2.3 Hough Edge Detector . . . . .	20
3.2.4 Relaxation . . . . .	21
3.2.5 Linking . . . . .	22
3.3 Point Features . . . . .	28
3.3.1 Intersections . . . . .	29
3.3.2 High Curvature Points . . . . .	30
3.3.3 Other Point Features . . . . .	33
3.4 Region Features . . . . .	36
3.4.1 Invariant Moments . . . . .	36
3.4.2 Region Segmentation . . . . .	38
4. Registration . . . . .	40
4.1 Correlation-Based Techniques . . . . .	42
4.1.1 Basic Correlation . . . . .	43
4.1.1.1 Correlating Abstract Edge Lists . . . . .	45
4.1.1.2 Correlating Abstract Triangle Lists . . . . .	47
4.1.2 Sequential Matching Methods . . . . .	49
4.1.3 Hierarchical Scene Matching . . . . .	51
4.2 Basic L.N.K. Registration Technique . . . . .	52
4.2.1 Example . . . . .	56
4.2.2 Full RST Transformation . . . . .	61
4.2.2.1 Example with Scale . . . . .	66
4.2.3 Extending LNK Registration to 3 Dimensions . . . . .	72
4.2.4 Clustering . . . . .	82
4.3 Region Matching . . . . .	86
4.3.1 Region Image Matching Using Similarity of Region Features . . . . .	88
4.3.2 Region Adjacency Graph . . . . .	90
4.4 Comparison of Registration Procedures . . . . .	93
4.5 Image Disparity Determination . . . . .	96
5. Conclusions and Recommendations . . . . .	98
5.1 Conclusions . . . . .	99
5.2 Recommendations . . . . .	101
6. References . . . . .	105
Appendix A. The L.N.K. Registration Procedure Software . . . . .	108

## 1. Introduction and Summary

The matching or overlaying of images from similar or dissimilar sensors is termed "registration". Matching of images with other images including symbolic images such as maps is a general and important problem in a wide variety of image analysis tasks, in particular in those aimed at extracting mapping, charting and geodetic data. The registration of digital and digitized images and digital maps is necessary in planned systems for creating and updating digital cartographic data bases such as the Digital Landmass System (DLMS) database.

Registration is distinguished from rectification which refers to the attempted correction of geometric distortions introduced during the acquisition of the imagery by the sensing mechanism and a variety of other sources which are categorized and briefly described in section 2.

As noted in section 2, not all distortions are correctable without registering the imagery with a database. This interdependence between rectification and registration together with the different resolutions, sensing geometries and different appearances of contents of imagery acquired by different sensors makes registration a difficult problem. In general, however, most imagery of interest will depict distinguishing structural characteristics such as lines, shapes, or tones, the aggregate of which allows the identification and analysis of the scene. While the individual structural features may be interpreted ambiguously the global geometric arrangement often admits an unambiguous interpretation for the scene and its various regions.

Individual points in an image field cannot always be uniquely interpreted. Points which are not uniquely identifiable by local structure and global geometry are called ambiguous points. A pass point is a point in an image identifiable by the structure of its local neighborhood and its geometric relation-



ship to other pass points and identifiable structures. Examples are forks in streams, mountain peaks, or road intersections. Points in wheat fields or lakes are not usually pass points. Pass points whose absolute location, i.e., latitude, longitude and elevation, on the globe is known are called ground control points.

Fortunately there are usually enough pass points evident in reasonably rich imagery to permit global matching of images even without ground control or platform attitude information. Automatic procedures for global matching based on pass point determination must however confront the fact that pass points are very sparsely distributed and the bulk of the image points are ambiguous points.

This document reports on our study of the problem of automatic registration of images with maps and images from dissimilar sensors. The registration problem is divided into four subproblems: (a) detecting and extracting appropriate features for pass point determination; (b) determination of an approximate global registration; (c) determination of disparity for a subset of points in the images; (d) determination of the global nonlinear transformation for the entire image.

For all methods other than the standard correlation of gray-scale imagery, the inexpensive extraction of good features is a primary concern. The cost of feature extraction must be balanced against the potential use of the features. Region features, while often costly to obtain, can be vital in later phases of image interpretation. On the other hand, isolated interest points, such as points of high curvature, can be inexpensive to obtain but may serve no purpose other than to determine registration.

Following a brief discussion on rectification in section 2, section 3 considers the detection and extraction of edge features, point features and region features. Edge detectors based on gradient and Laplacian operators and Hough

transforms are described. A simple linking method which produces reasonably long edge segments and produces good results is also presented. Point feature detectors include two point operators and detectors for intersections and high curvature points. Region features based on invariant moments and region segmentation are described. This section also presents results of applying some of the feature detectors to a panchromatic image and infrared image of the same scene.

Section 4 is devoted to approaches to registration and includes a brief survey of some existing techniques for matching. Section 4.1 describes the basic correlation method, a sequential correlation method, a hierarchical correlation method, presents example of correlating feature vectors where the features are edges, and another example using abstract triangles formed by connecting three successive high curvature points. Section 4.2 presents the basic L.N.K. registration procedure and examples of its application to images. Included is a brief report on an experiment with an example involving scale change. The experiment demonstrates the utility of the proposed technique for regions where cultural activity creates features such as straight edges or networks of lineals. The results reported are typical of many similar experiments.

Section 4.2.3 presents an extension of the LNK registration method to 3-dimensional images. Good approximate rotation, scale and translation (RS&T) transformations were obtained automatically for photo/map pairs even when there was some relief in the terrain. However there are many cases, such as in low altitude aerial imaging, where RS&T transformations are inadequate. In such cases projective transformations must be used and this section shows how the LNK registration technique can be extended to 3-d under certain constraints.

All versions of the L.N.K. registration technique involve clustering in parameter space. Section 4.2.4 discusses two techniques - hierarchical cluster-

ing and variable resolution clustering - which L.N.K has used in its registration procedure.

Symbolic matching of images consists of reducing two images to a pair of abstract structures and comparing these structures. Section 4.3 considers methods which take the regions in a segmented image as components of the structure describing the image. Most of the available region matching algorithms do not provide the registration transformation but only a correspondence between some subset of the set of regions in the first image and a subset of the set of regions in the second image. Section 4.3.1 describes region image matching using similarity of region features and section 4.3.2 considers an approach in which graph theoretic techniques are used to match region segmented images represented as graphs. Each node of the graph corresponds to a region and an edge in the graph represents adjacency between two regions. In this Region Adjacency Graph (RAG) each node also contains a set of descriptors such as shape, size, and average brightness, about the region represented.

Section 4.4 presents a comparison of registration procedures. Because of the unavailability of relevant imagery, the scope of the contract and the less than precise descriptions available in the publications describing various methods, it was feasible to present only a brief qualitative comparison.

Once a global registration has been determined it is usually necessary to modify the transformation to account for local distortions. The disparity between two registered images is the small local differences in corresponding pixel locations caused by these distortions. Section 4.5 discusses the determination of image disparity. Section 5 presents our conclusions and recommendations for further investigation.

## 2. Rectification

All remote sensing imagery have geometric distortions. These distortions can be grouped into six categories:

- (1) distortions caused by the topography,
- (2) distortions caused by the sensing mechanism,
- (3) distortions caused by the recording mechanism,
- (4) distortions caused by a non-ideal flight path,
- (5) distortions caused by perspective,
- and (6) distortion caused by movement of the earth.

All these distortions would make registration more difficult. Thus, all correctable distortions should be removed before applying any registration techniques.

Distortions due to topography include changes in position caused by the terrain's altitude and distortions caused by the earth's curvature. In principle, the distortions due to altitude can be eliminated if there is an altitude database available for that region. The only problem with using it, is that the image must be registered first. Because of this conflict, registration techniques should be flexible enough to allow for topographical distortions.

If the distortions caused by the sensing and recording mechanism can be modelled, then it may be possible to completely remove the distortion. In most cases, some of the parameters of the model are known approximately. In this case only some of the distortion can be removed.

For example, tangential distortion of line scanner imagery, in which the scale varies across the flight path, causes lineal features across the flight path to have an "S" shape. In order to precisely determine the necessary correction for a particular pixel, the height of the sensor above that pixel needs to be known. As mentioned before, even with an available altitude database,

the image must be registered before it can be used. For the above example, the tangential distortion is removed by using an approximate height above the terrain which is usually the average altitude of the aircraft carrying the sensor.

Distortions due to non-ideal flight path include those due to any deviation from the optimal orientation of the aircraft and distortions due to movement of the aircraft during image taking.

Distortions due to non-optimal orientation share the problem that their corrections need the terrain's altitude. Correction requires the parameters of the flight path to be known.

Distortions due to perspective occur, in addition, if the orientation of the sensor is not vertical or the field-of-view of the sensor is large. Precise correction, again, requires the terrain's altitude.

Finally, movement of the earth can be a problem in satellite scanner imagery since it causes the image to have a shape like a parallelogram.

### 3. Feature Selection

#### 3.1 Feature Detection

The role of feature selection in registration is an important one. Each registration method requires a certain image representation, which contains a set of features representing the image. This in turn, means a set of feature detectors to detect those features is needed. No registration technique can be better than its corresponding feature detectors. If the method requires edges and the edge detector worked poorly, then the registration will not be good. Hence, what features can be detected cheaply and reliably will greatly influence the choice of a registration method.

This section is divided into three parts, edge features, point features, and region features. In order to show the utility of some of the features presented, the feature detectors were applied to a panchromatic image and infrared image of the same scene. The results were then correlated and examined. The correlation techniques used were basic ones and are later discussed in section 4.1.1.

### 3.2 Edge Detection

There is evidence that registration of two images should be based on the edge information in the images rather than directly matching pixels in the images using their grey levels. Attempts to match using grey level values have not been very successful. Crombie [1975] has reported anomalous matching when using correlation techniques applied directly to grey-scale images. Horn and Bachman [1977] used a hill climbing technique on grey scale values, but their method requires a good initial approximate registration and is also computationally expensive. In addition, these problems are compounded when registration of images from different sensors is attempted. For example, light-dark reversal is possible when comparing a panchromatic image with an infrared image of the same scene. Even more of a problem, are radar images that may have totally black shadows where a panchromatic image of the same scene shows considerable texture.

One argument for edge matching is based on results of experiments with human beings. People seem to differentiate regions by using the regions' boundaries. John Krauskopf of Bell Laboratories [Gilchrist 1979] performed experiments on human beings and discovered they identify the color of the interior of a region using contrast information at the boundary edges of the region. When the boundary is made to disappear, by artificially stabilizing the image, the perceived color of the region changes to the color of the surrounding region.

Another, more practical argument, is that edge-based registration of both images to maps and of images from different sensors, has been successfully performed. Savol [1978] matched radar images with maps using edges. Hall [1979] matched panchromatic images with radar images using correlation

techniques applied to edges.

This section deals with various edge based feature detectors. The output of the edge detectors is a binary image whose pixels with value one, are the corresponding edge points in the original image. The usefulness of edges for registration is demonstrated by testing the results of some of the edge detection techniques. This was accomplished by applying the edge detectors to two dissimilar images of the same scene and matching the resultant edge images using the simple correlation technique. Although the application used two images, it just as easily could have been done using a map.



### 3.2.1 First Derivative Methods

One simple and very common method of finding edge points in an image is to apply a threshold to the magnitude of the gradient (first derivative) of the image. There are many ways of approximating the gradient [Davis 1975] using small window sizes. Small window sized methods are desirable because they are computationally fast; some methods have been hardware implemented. Unfortunately, they also are vulnerable to noise points. More reliable edge points can be found by using larger windows [Rosenfeld and Thurston, 1970; Hayes and Rosenfeld, 1970] but the ease of computation decreases markedly. In all cases the choice of the appropriate threshold value has to be determined.

An example of a small-window approximation to the gradient is the Sobel detector which is defined using the 3x3 neighborhood of a pixel as shown in Figure 3-1.

A	B	C
D	E	F
G	H	I

Figure 3-1. 3x3 Neighborhood of Pixel E.

The Sobel magnitude value for pixel E is found by

$$|A + 2B + C - G - 2H - I| + |A + 2D + G - C - 2F - I|.$$

The Sobel operator was applied to two images of the same scene but taken from different sensors. One image was a 1024 x 1024 panchromatic image of a scene of the Canadian border (CBP52C). The other image was the corresponding infrared image (CBI52C). The lower left 512 x 512 quarter of the panchromatic image will be referred to as image BW-CB. The corresponding quarter of the infrared image will be referred to as IR-CB.

Using a threshold of 166, the Sobel operator found 44,536 edge points (17%) in image BW-CB, and 16% in image IR-CB. The central 256 x 256 region of detected edge points from the images were correlated and part of the correlation matrix is shown in Figure 3-2. The values of the rest of the matrix decrease from those shown. The correlation peak occurs at the displacement ( $x = -4$ ,  $y = 2$ ).

	-5	-4	-3	-2	-1
-1	1374	1321	1274	1245	1237
0	1410	1427	1398	1331	1277
+1	1281	1531	1477	1422	1388
+2	1564	1623	1566	1482	1410
+3	1547	1595	1565	1527	1471
+4	1469	1464	1492	1402	1390

Figure 3-2. A portion of the correlation matrix of the central regions of Sobel edge points from images BW-CB and IR-CB.

The experiment was repeated using a threshold value of 260. In this case, 7% of BW-CB pixels and 6% of IR-CB pixels were identified as edge points. Unfortunately, the correlation produced five peaks. Two high peaks occurred at displacements (3, -5) and (-5, -3), while there were three other peaks at (-1, 1), (0, 3), and (5, 3). It appears that setting the threshold too high will cause noise points to be preferentially selected making correspondence unreliable. Since the correct transformation is known to be ( $x=0$ ,  $y=0$ ), the above experiments show that a fairly good registration transformation can be found just using a simple edge detector.

### 3.2.2 Marr Edge Detector

Marr [1979] used the Laplacian of a two-dimensional Gaussian distribution convolved with an image as an edge detector. The Laplacian is an approximation to the second derivative of the image. It is parameterized by a variable,  $\sigma$ , so that the Laplacian can be made smooth and band-limited in the frequency domain, and smooth and localized in the spatial domain. By varying  $\sigma$ , objects within a determined size range can be detected and pinpointed in the image.

The two dimensional Gaussian is defined as

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

and its Laplacian is

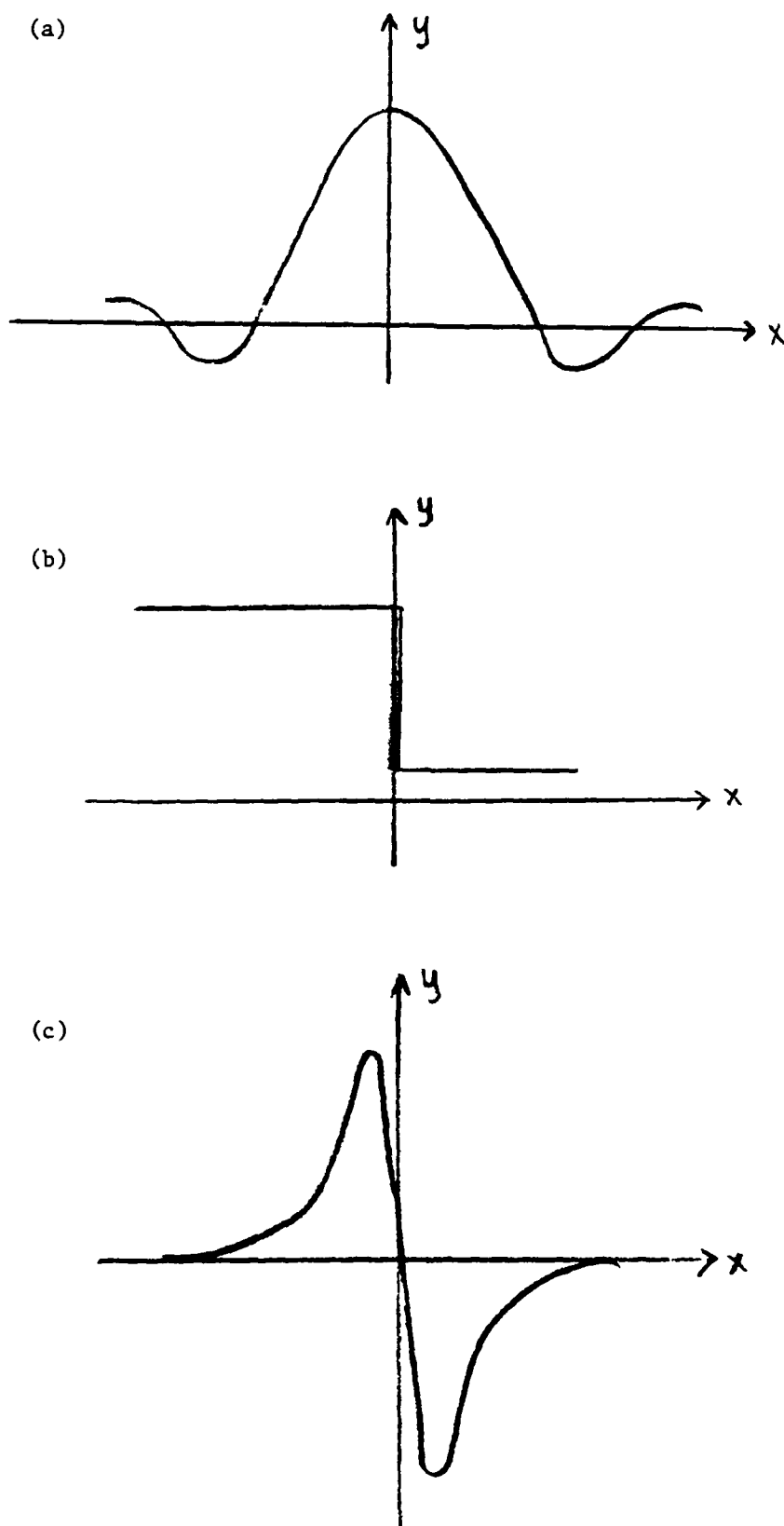
$$L(x,y) = G''(x,y) = \frac{1}{2\pi\sigma^6} (x^2 + y^2 - \sigma^2) e^{-\frac{(x^2+y^2)}{2\sigma^2}}.$$

A cross-section of  $L(x,y)$  is shown in figure 3-3(a). The two-dimensional  $L(x,y)$  is found by rotating the graphed curve about the y-axis. The central portion of  $L(x,y)$  is negative and is  $2\sigma$  wide. The outer portion of  $L(x,y)$  is positive. The peaks of  $L(x,y)$  occur at  $x = \pm\sqrt{3}\sigma$  from the origin.

A cross-section of an edge is shown in figure 3-3(b). When  $L(x,y)$  is convolved with this edge, the curve in figure 3-3(c) results. Note that where this curve crosses the x-axis is where the edge occurred in figure 3-3(b). Therefore edges are found by locating the zero-crossings of the Laplacian convolved with the original image.

One interesting property of the Marr edge detector is it tends to find continuous (gap-free) edge segments. Most other edge detection methods do not do this, and require additional processing to weed out

Figure 3-3. The Marr edge detector (a) convolved with an edge (b) produced (c), where the zero crossing indicates an edge.



noise points and fill in missing pieces. The continuous edge segments can then be easily represented as chain codes [Freeman 1974].

The Marr edge detector with  $\sigma=5.5$  was applied to the BW-CB and IR-CB images. The outer positive portion of the mask was chopped off after six values making the whole mask 23 pixels in radius. The detector classified 17% of both BW-CB and IR-CB image pixels as edge points. The central 256 x 256 portions of the two sets of edge points were correlated and part of the resulting correlation matrix is shown in figure 3-4. The correlation peak occurs at the displacement ( $x = -3$ ,  $y = 3$ ). Compared to the results of matching using the Sobel detector (figure 3-2), the Marr detector performed much better, as the Marr peak value has twice as many matches as the Sobel correlation peak value.

In order to allow for corresponding edges in two images to be off by a pixel, it may be desirable to thicken the detected edges [Hall 1979]. One way to do this is to assign a weight of 3 to all detected edge points, assign a weight of 2 to background points that are next to an edge point, and a weight of one to background points that are two pixels from an edge point. All remaining background points are assigned a weight of zero.

The thickened edge method was applied to the detected Marr edge points of both images, and the central 256 x 256 portion of the results were again correlated. The correlation matrix, in this case, is defined by

$$C(i,j) = \sum_{x=129}^{384} \sum_{y=129}^{384} T_1(x,y) * T_2(x+i, y+j),$$

where  $T_1(x,y)$  and  $T_2(x,y)$  are the values of the thickened edge images for BW-CB and IR-CB, respectively.

Y	X							
	-6	-5	-4	-3	-2	-1	0	1
-1	2216	2292	2246	2092	1935	1871	1895	2050
0	2394	2538	2559	2322	2059	1857	1783	1876
1	2460	2758	3000	2765	2348	2009	1838	1829
2	2441	2887	3356	3320	2793	2309	2003	1874
3	2107	2565	3234	3494	3127	2681	2336	2050
4	1849	1829	2162	3125	3096	2778	2475	2177
5	1739	1888	2144	2493	2664	2624	2488	2299

Figure 3-4. A portion of the correlation matrix of the central regions of Marr edge points from images BW-CB and IR-CB.

	-6	-5	-4	-3	-2	-1	0	1
-1	153406	155137	155376	153440	149975	146641	144793	145011
0	155619	158924	160258	158679	154401	149357	145670	144414
1	157801	162763	165849	165060	160588	154240	148745	145660
2	158032	164422	169255	170045	166171	159528	152816	148326
3	155399	162186	168342	170827	168507	162821	156103	151003
4	150849	156762	163092	166923	166528	162471	157167	152449
5	146542	150680	156006	160223	161319	159420	155768	152425

Figure 3-5. A portion of the correlation matrix of the central regions of thick Marr edge points from images BW-CB and IR-CB.

The center portion of the correlation matrix is shown in figure 3-5. The peak value occurs at displacement ( $x = -3$ ,  $y = 3$ ) is the same as the result obtained from the unthickened case. However, the peak value in the thickened correlation matrix is much stronger (since  $170827/3 > 17089$ ). These experiments again show that registration using edges seems to work well.

It should be mentioned that the Marr detector at  $\sigma=5.5$  picked up texture elements within the fields and woods in addition to the boundaries between woods and fields. At a higher value for  $\sigma$ , larger regions would be picked out. Figure 3-6 shows a panchromatic image and Figures 3-7(a)-(c) show the results of applying the Marr detector with different  $\sigma$  values.





Figure 3-6. Image used to demonstrate  
Marr edge detector.



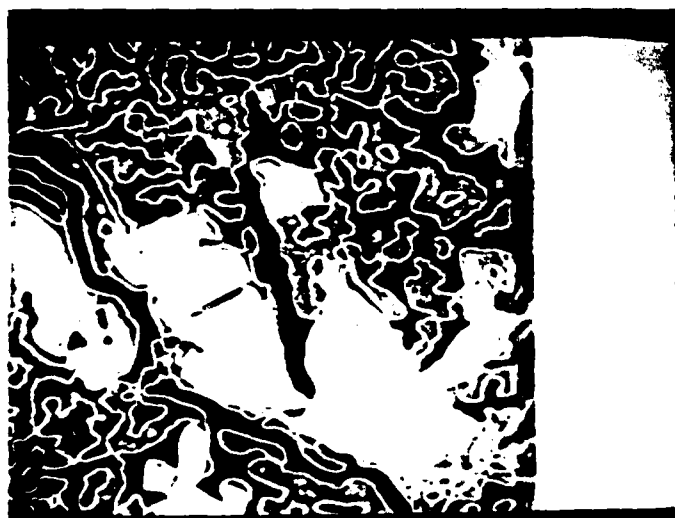
(a)

Figure 3-7. Results of applying Marr edge detector to image in Figure 3-6 with,

- (a)  $\sigma=2.5$
- (b)  $\sigma=5.5$
- (c)  $\sigma=8.5$ .



(b)



(c)

### 3.2.3 Hough Edge Detector

The Hough transformation is an efficient device for detecting if a set of high contrast points are organized along a mathematical curve [Duda and Hart, 1972]. The simplest mathematical curve and the most important one for detection of man made structures is the straight line. Only two parameters are required for specification of a given line--in polar form the parameters are the direction of the normal to the line,  $\theta$ , and the distance from the origin to the line,  $r$ . If the possible line directions are discretized to  $T$  values and the possible distances from the origin are discretized to  $R$  values then Hough detection is logically equivalent to a matching of  $T$ - $R$  templates to the high gradient points [Stockman and Agrawala, 1977]. The Hough straight line detector is an excellent device for detection of man made structures in imagery as proven by various L.N.K. experiments on aerial imagery.

There are also Hough detectors for circles, parabolas, ellipses, and hyperbolas. LNK has performed some experiments in registration using the Hough circle detector, [Stockman and Kopstein, 1979]. The other Hough detectors tend not to be as efficient as the Hough line and circle detectors, because they require estimation of too many parameters.

Examples of the result of applying the Hough edge detector to panchromatic images are shown in the LNK registration procedure section.

#### 3.2.4 Relaxation

It is possible to apply the technique of relaxation to edge detection [Zucker 1977]. All possible edges are divided into groups according to their orientations. For each group, the probability that a pixel belongs to an edge in that group is calculated, plus the probability that the pixel belonged to no edge. A label consisting of the above set of probabilities is assigned to each pixel. Then, through a parallel iterative process, the labels of adjacent pixels are compared and compatible orientations cause their probabilities to strengthen while incompatible orientations cause their probabilities to weaken. The process stops when all the probabilities in all the labels have approached their limiting values. The initial probabilities can be found by using the simple edge detectors discussed in section 3.2.1.

The detected edges can then be represented by a chain code. The relaxation procedure tends to thicken edges, so additional processing may be necessary to compensate for this.

### 3.2.5. Linking

Most of the edge detection methods described so far produce only edge points (Sobel detector) or edge segments (Hough detector). While the Marr detector can be made to produce relatively long edge segments it is computationally expensive. It is possible to do additional processing on the simpler methods to try to produce longer edge segments.

One such method takes three steps. The first step is to extract all high contrast pixels. A gradient operator is then applied to each high contrast point to determine the gradient's magnitude and direction at the point. In the second step, a small neighborhood about each high contrast point is spirally examined to find the best (if any) continuing high contrast point in the forward and backward direction. If such points exist, links are established to them. Finally in the last step, the edge segments are extracted from the chained high contrast points. All chains less than a certain length can be eliminated as noise edges. Note, the linking step can be done in parallel for each high contrast point.

LNK has applied this method and achieved good results by keeping only the top 5% of high contrast points. Figure 3-8(a) shows a light airplane on a darker airfield. The result of applying step one to a window containing the right wing tip, i.e. finding the high contrast points and applying the gradient operator at each point is shown in Figure 3-8(b). The result of applying the second step of linking is shown in Figure 3-8(c). Notice that some of the points are of degree three, i.e. they are at the junction of multiple edge activity. These points must be at the terminus of an extracted curve segment because they cannot relate symmetrically to three neighbors. The result of applying the last step is shown in Figure 3-8(d). A large portion of the wing

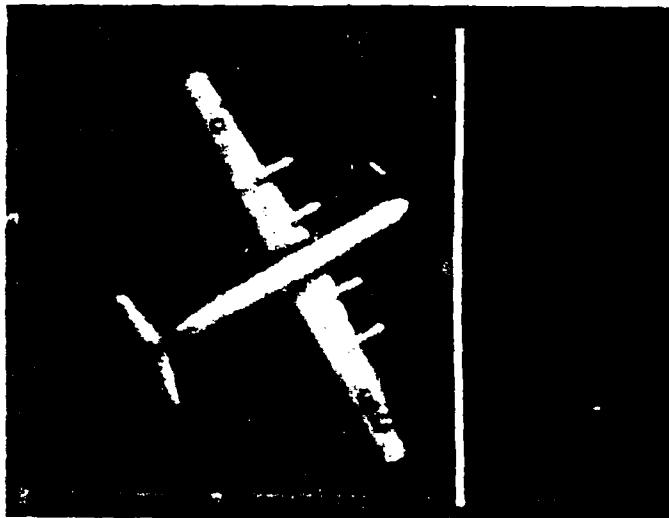


Figure 3-8(a). Airplane on airfield background.

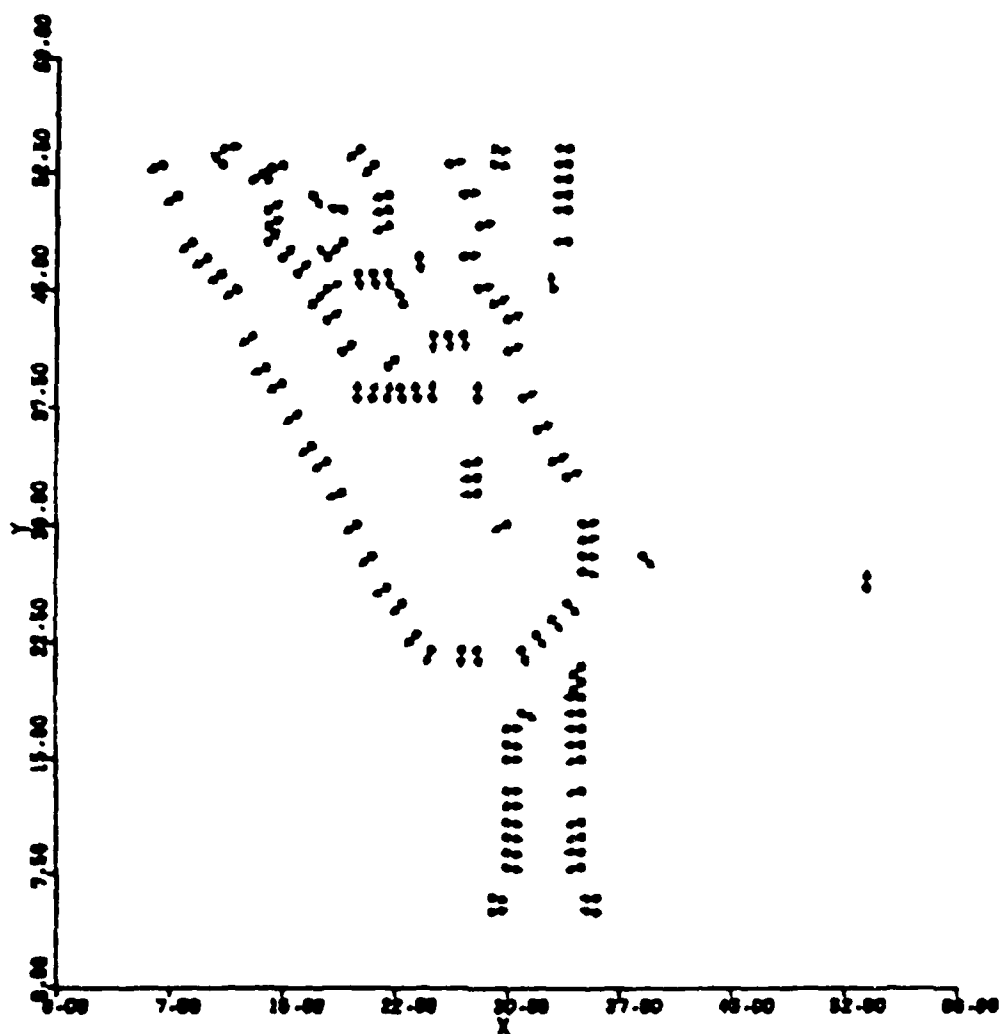


Figure 3-8(b). Gradient direction of high contrast points of right airplane wing.  
(Curve detection step 1.)

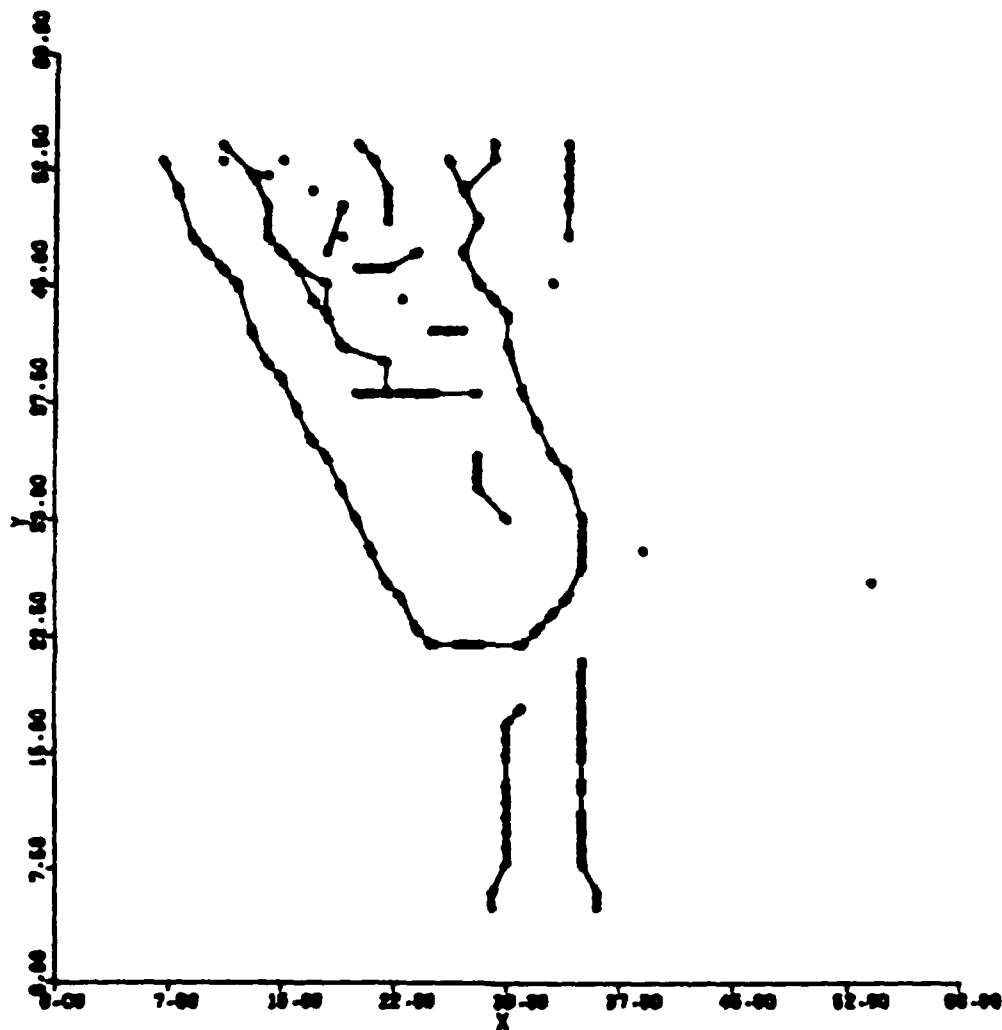


Figure 3-8(c). Plot of all forward and backward linking relationships among high contrast points of (b). (Curve detection step 2.)



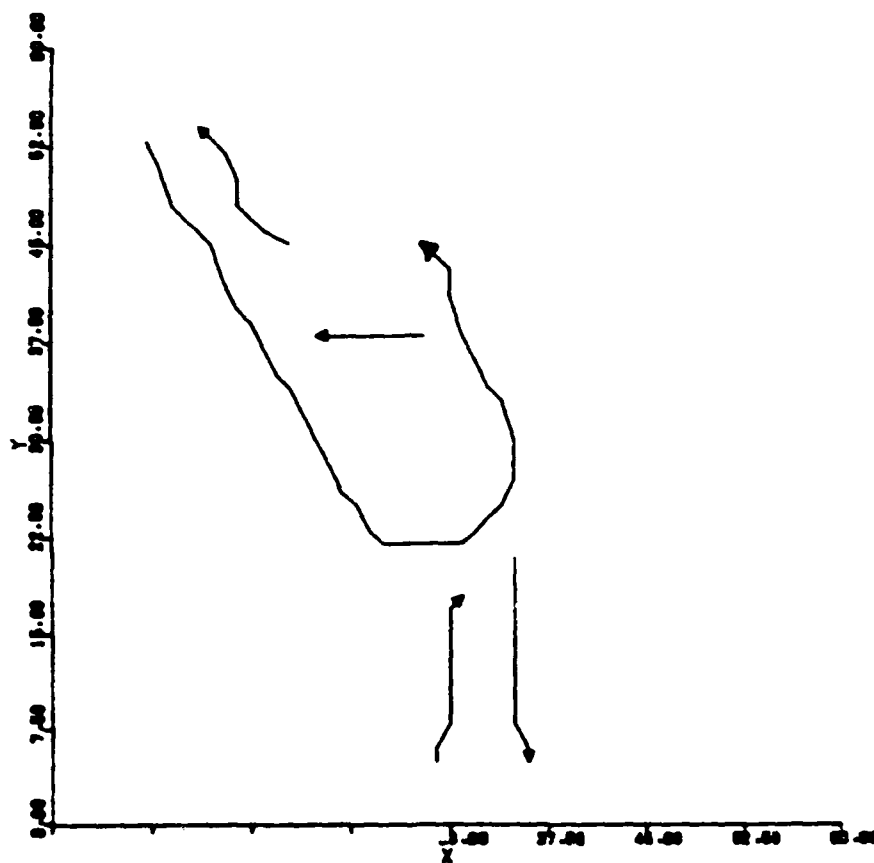


Figure 3-8(d). Long curve segments extracted  
from related points of (c).  
(Curve detection step 3.)

boundary was successfully extracted along with two edges of the "USAF" identification interior to the wing plus two edges of a dark streak on the airfield below the plane.

The linking of edge points or short edge segments can be much more complicated. Higher level processing using model-directed techniques can be used to make linking decisions. Geometric or topological constraints can also be used. These type of methods produce only a very restricted set of edges and thus are not very general.

### 3.3 Point Features

Point features can also be used for matching and in some ways, may be more desirable than edge features. One problem with edge features, is that the length of the detected edge segments is likely to vary among images of the same scene. Since it is difficult to tell which parts of the edges correspond, many edge matching algorithms use the end points of the edges. But, if the endpoints of the detected edges are likely to vary, this makes accurate registration difficult. Point features, on the other hand, do not have this problem.

Point features need not be the result of some point detectors. For example, the intersection of two lines or edges determines a point. Such point features are, in a sense, a higher order feature than those derived from a point detector. This section discusses two higher order point features, intersections and high curvature points, and two point detectors, the Moravec interest operator and a "special" point operator. In later sections, methods of forming edges by connecting point features and using these for registration, are discussed.

### 3.3.1 Intersections

Intersections between two or more lines or edges can provide a good base for registration. The topology and geometry of intersections can be used to classify intersections into various types. For example the number of lines or edges in the intersection, the angles involved, and the relative grey levels of the regions between the lines or edges can all be used to type an intersection. The typed intersections can then be used to aid matching.

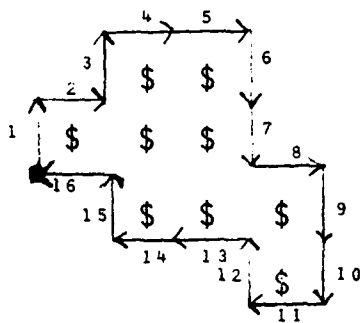
Since most edge detectors tend to be unstable at intersections, some additional processing will need to be done to force the intersections to occur. It is even possible to create imaginary intersections as the surveyor does. For instance, the wall of a building can be extended to the street to form an intersection. Matching using intersections has been studied [Zahn, 1974; Dudani, 1977; Stockman and Kopstein, 1979].

### 3.3.2 High Curvature Points

Not all images will contain intersections. Intersections tend to stem from man-made structures such as road networks, cultivated fields, or buildings. In images with no man-made features, intersections are less likely to occur. In such images, high curvature points on curved edges can be used as registration features instead.

Once long continuous edges have been found, they easily can be represented by chain codes or crack codes. Figure 3-9(a) shows a small region formed by '\$'s and the links along the "cracks" of the region. Figure 3-9(b) shows how these links can be represented by a code. The crack code for the links of the region in Figure 3-9(a) is

0 1 0 1 1 2 2 1 2 2 3 0 3 3 0 3.



(a)

↑	0
→	1
↓	2
←	3

LINK CHAIN CODE

(b)

Figure 3-9: Crack Code Representation

The high curvature points of a curve represented by a crack code can be defined as follows. Let the l-curvature of a point P on a curve be defined as the angle between the line, connecting the previous point on the curve and P, and the line, connecting P and the next point on the curve. The K-curvature

can then be defined as the angle formed by the line, connecting the  $K^{\text{th}}$  previous point and P, and the line connecting P and the  $K^{\text{th}}$  succeeding point. In general, the curvature measures from  $-179^{\circ}$  to  $+179^{\circ}$  with positive values indicating the curve is bending clockwise.

For example, Figure 3-10 shows a portion of the crack code of a curve. R is the sixth point before point P and Q is the sixth point after point P. The K-curvature for point P, for  $K=6$ , is the angle  $\alpha$  shown, which is  $90^{\circ}$ .

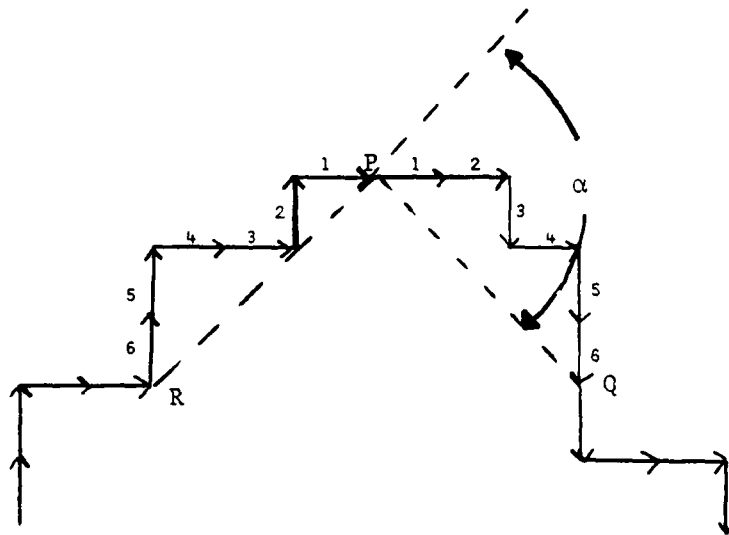


Figure 3-10. An example of K-curvature for point P where K is 6. The K-curvature is  $\alpha$  which is  $90^{\circ}$ .

A technique called non-maximum suppression can be used to isolate the high curvature points. The curvature of each point is compared with its K neighbors on both sides. If the curvature of the point is more positive or more negative than all of its  $2K$  neighbors, then it is a high curvature point if its absolute curvature is greater than  $90^{\circ}$ .

This technique was applied to the Marr edges, extracted from the images BW-CB and IR-CB, discussed in Section 3.2.2, with  $K=10$ . Since a  $\sigma=5.5$  was used, the Marr detector picked out many small regions (see section 3.2.2) so that there were many high curvature points. A higher value for  $\sigma$  would cause

larger regions and fewer high curvature points; thus a higher value for K would be needed.

The high curvature points of the two images were correlated and part of the correlation matrix is shown in Figure 3-11. The correlation peak occurred at displacement ( $x=-2$ ,  $y=0$ ) which is closer to the correct transformation than the ones obtained just using the edges. However, in this case, the full 512 x 512 region was correlated instead of the central 256 x 256 region as before and this may be the cause of the difference.

		X								
		-5	-4	-3	-2	-1	0	1	2	3
Y	-3	8	10	6	9	12	11	16	8	11
	-2	15	11	8	11	12	14	13	9	8
	-1	34	35	35	42	34	36	30	26	21
	0	78	81	91	<u>106</u>	82	74	67	62	64
	1	36	42	46	55	40	36	27	21	21
	2	24	38	46	53	43	37	30	23	20
	3	10	15	24	29	33	35	23	13	9

Figure 3-11. A portion of the correlation matrix of high curvature points from the images BW-CB and IR-CB.

The cluster in figure 3-11 is elongated along the x-axis. This is because there are several ragged horizontally-oriented edges in the images. The elongation could then have been caused by mismatching a high curvature point with a neighbor of the corresponding high curvature point in the other image. All registration techniques have problems matching regular patterns although what is considered regular depends upon the technique involved.

### 3.3.3 Other point features

Moravec [1977] proposed an "Interest Operator" as a point feature detector. The value of the interest operator is the minimum value of the variance of the gray scale values, computed in four directions (horizontal, vertical, and the two diagonals) over a small window.

The interest operator is especially sensitive to corners, spots and unfortunately, noise. The operator, of a particular window size, would be applied to the whole image and points whose interest value were relatively high would be selected. Care in picking the threshold would be needed so that noisy points are not the only points chosen.

It is possible to define a set of "special" points for each image by finding the "special" point in each row, column, and diagonal in the image. These special points are found by applying any long one-dimensional detector to each row, column, and diagonal, and selecting the point with the highest detector response.

There are several advantages to this method. The first is that, only a few points will be selected which cuts down the computation of matching. Second, the points are automatically distributed at least somewhat throughout the image, since each row, column, and diagonal will have a special point. This method would work best, either matching images from the same sensor, or matching images to maps. Note, the map need only store a small number of points.

The one dimensional detector could be an edge detector, line detector, texture detector, and so on. An experiment using a one-dimensional edge detector was performed on images BW-CB and IR-CB, described in Section 3.2.1. For each row and column, (the diagonals were not used), the point which had the greatest difference between the sum of the preceding 50 points and the sum of the fol-



lowing 50 points, was chosen as the special point. The special points were correlated and a portion of the correlation matrix is shown in Figure 3-12. The cluster is again elongated along the horizontal as in Figure 3-11 and probably for the same reason, i.e. mismatching of neighboring special points along a prominent edge. The peak occurs at  $(x=-3, y=0)$  which is different from those obtained before.

Y	X																				
	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10
-10	6.	8.	14.	7.	17.	13.	19.	12.	13.	9.	15.	6.	8.	10.	9.	9.	6.	6.	6.	8.	3.
-9	7.	11.	12.	6.	12.	15.	10.	19.	14.	14.	14.	10.	10.	16.	13.	11.	8.	5.	9.	3.	4.
-8	8.	5.	16.	10.	18.	14.	10.	11.	19.	14.	12.	10.	23.	14.	13.	6.	8.	10.	12.	9.	7.
-7	6.	7.	12.	10.	14.	17.	16.	12.	15.	14.	17.	7.	15.	11.	9.	15.	6.	9.	12.	11.	5.
-6	6.	13.	8.	14.	18.	20.	16.	21.	18.	17.	17.	11.	16.	16.	12.	12.	8.	10.	12.	2.	2.
-5	7.	7.	6.	16.	16.	20.	24.	17.	19.	20.	14.	12.	16.	10.	13.	10.	15.	11.	12.	4.	7.
-4	6.	15.	3.	15.	18.	23.	18.	20.	20.	20.	10.	10.	15.	14.	11.	13.	18.	8.	12.	7.	10.
-3	3.	8.	8.	14.	14.	10.	29.	21.	22.	20.	19.	12.	11.	15.	8.	14.	10.	12.	13.	10.	13.
-2	8.	9.	10.	10.	17.	13.	12.	27.	26.	19.	26.	14.	16.	17.	13.	15.	12.	15.	15.	21.	9.
-1	10.	8.	16.	19.	24.	15.	26.	25.	34.	33.	26.	19.	23.	11.	18.	12.	16.	11.	22.	14.	15.
0	32.	37.	39.	43.	48.	44.	49.	61.	53.	46.	53.	56.	44.	33.	36.	40.	34.	42.	39.	36.	35.
+1	5.	8.	15.	10.	20.	24.	38.	22.	32.	36.	31.	25.	19.	11.	13.	14.	9.	10.	12.	8.	7.
+2	7.	8.	7.	11.	11.	18.	27.	30.	44.	32.	44.	37.	19.	20.	12.	16.	16.	11.	7.	14.	7.
+3	7.	5.	9.	10.	11.	22.	27.	30.	38.	23.	37.	32.	25.	17.	15.	17.	9.	9.	12.	8.	11.
+4	2.	4.	8.	11.	15.	13.	22.	24.	22.	39.	48.	29.	19.	19.	15.	16.	10.	11.	14.	12.	10.
+5	9.	6.	7.	9.	9.	18.	17.	13.	20.	20.	36.	30.	24.	19.	14.	10.	10.	13.	15.	12.	12.
+6	10.	11.	14.	9.	10.	11.	10.	17.	17.	24.	33.	24.	25.	17.	16.	7.	11.	9.	13.	19.	13.
+7	7.	9.	5.	8.	7.	14.	6.	17.	11.	18.	32.	22.	21.	17.	19.	15.	10.	6.	12.	10.	9.
+8	11.	10.	9.	5.	11.	7.	8.	9.	12.	21.	23.	23.	19.	18.	8.	11.	11.	11.	8.	7.	6.
+9	7.	6.	7.	11.	10.	8.	13.	6.	5.	13.	24.	15.	20.	20.	16.	10.	6.	12.	13.	9.	5.
+10	10.	3.	8.	13.	4.	9.	10.	13.	10.	12.	19.	11.	17.	21.	16.	9.	14.	8.	13.	10.	2.

Figure 3-12. Part of the correlation matrix of special points from images BW-CB and IR-CB.

### 3.4 Region Features

#### 3.4.1 Invariant Moments

Features on images or parts of images which are invariant under rotation, translation, and scale changes can be used for scene matching. It is possible to define such invariant features using the moments of the image gray scale function. The problems with moments are (1) digitization may cause the moments to no longer be invariant and (2) finding an easily computable set of moments that can adequately characterize an image.

Wong and Hall [1978] have devised a set of seven invariant moments which they claim are adequate for matching two images. Let the gray scale function be denoted  $f(x,y)$ , then the  $pq^{th}$  central moment of the image is defined by

$$N_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y),$$

where  $\bar{x}$  and  $\bar{y}$  are the means of  $x$  and  $y$ , respectively, over the whole image.

The normalized central moments  $pq$  can then be defined by

$$B_{pq} = \frac{N_{pq}}{N_{00}^y},$$

where  $y = (p + q)/2$ .

The set of seven moments which are invariant under translation, rotation, and scale are then defined by

$$M_1 = B_{20} + B_{02},$$

$$M_2 = (B_{20} - B_{02})^2 + 4B_{11}^2,$$

$$M_3 = (B_{30} - 3B_{12})^2 + (3B_{21} + B_{03})^2,$$

$$M_4 = (B_{30} + B_{12})^2 + (B_{21} + B_{03})^2,$$

$$M_5 = (B_{30} + 3B_{12})(B_{30} - B_{12})[(B_{30} + B_{12})^2 - 3(B_{21} + B_{03})^2]$$

$$+ (3B_{21} - B_{03})(B_{21} + B_{03})[3(B_{30} + B_{12})^2 - (B_{21} + B_{03})^2]$$

$$M_6 = (B_{20} - B_{02})[(B_{30} + B_{12})^2 - (B_{21} + B_{03})^2] + 4B_{11}(B_{30} + B_{12})(B_{21} + B_{03}),$$

$$M_7 = (3B_{12} - 3B_{30})(B_{30} + B_{12})[(B_{30} + B_{12})^2 - 3(B_{21} + B_{03})^2] + (3B_{21} - B_{03})x(B_{21} + B_{03})[3(B_{30} + B_{12})^2 - (B_{21} + B_{03})^2].$$

There are many possible ways of using the invariant moments. The original image could be subdivided into smaller subimages or regions and each subimage or region could be represented by its own set of invariant moments, and these could be used to correlate with other images or a map. Section 3.4.2 presents some methods of region segmentation and section 4.1.3 discusses a method of finding subimages.

### 3.4.2 Region Segmentation

A segmentation of an image is a partition of the image into disjoint subsets whose union is the entire image. For surveys of segmentation see [Riseman and Arbib 1977, Kanade 1980, Zucker 1976, Pavlidis 1977]. Many segmentation procedures produce an initial segmentation and then apply an iterative procedure such as merging or splitting to obtain an improved segmentation. Two basic approaches to finding an initial segmentation are first locating boundaries or first locating pixels with similar feature values.

Boundary detection generally consists of edge detection followed by linking of edges into closed boundaries as discussed in section 3.2.5. Edge detection procedures which form closed edges such as the Marr detector (sec. 3.2.2) and relaxation labelling (section 3.2.4) provide a segmentation directly. Edge detection in textured images is a difficult task which depends heavily on the relative scale of the texture and the regions' sizes.

Construction of regions based on similarity of pixels or pixel neighborhoods requires feature extraction. Features commonly measured include average gray level in a neighborhood, variance, average edge content per unit area, average orientation of local edges and average spot size of uniform contiguous areas. Texture measures, such as co-occurrence matrices and Fourier transforming data can be used as feature measures for larger areas. Threshold techniques [Pavlidis 1977, Price 1976] can be used with these features to provide an initial segmentation.

Milgram [1978] describes a procedure for region construction using evidence from several sources such as edge information and pixel feature values. The algorithm selects the contours at different thresholds according to the support of the edge data along the contours. Zucker [1979] gives a relaxation technique for constructing regions from primitive edges. This scheme

allows points to be considered as interior or boundary points of a region. The relaxation process allows edge segments separating regions to prosper as region points and edge points reinforce themselves.

Pavlidis describes a general class of split-merge algorithms using the notion of region adjacency. Given a criterion for deciding if a single region should be split and a criterion for deciding whether two adjacent regions should be merged, the procedure is as follows:

- 1) Split each region which should be split according to the splitting criterion. Continue this until no further regions satisfy the splitting criterion.
- 2) If step 3 has not been executed yet then keep going, else if no merges occurred in the last execution of step 3, then stop.
- 3) Merge any two adjacent regions which should be merged according to the merging criterion. Continue this step until no adjacent regions satisfy the merging criterion.

#### 4. Registration

The goal of registration is to find a correspondence of pixels between two images or between an image and a map. In this section we survey some existing techniques for matching. Before proceeding with the description of current methods we will attempt to give a general framework for discussing matching problems.

Matching involves a comparison of representations. We may classify matching algorithms by the type of representation we are comparing. The representation may be the original image, a hierarchy of images of varying sizes and resolutions, an image derived from the original image, a graph, or a list of points, edges or regions together with a set of descriptors. Derived images may consist of regions of constant intensity, region boundaries, edges or isolated points. Matching techniques depend upon the representation selected. Correlation techniques can be applied to images, graph isomorphism techniques may be applied to graphs, relaxation procedures may be applied to lists, and clustering may be applied to points, edges, and abstract vectors and triangles. A hierarchy of derived images may be used to locate subregions of an image which are likely to provide a quick accurate match.

The representations and matching procedures we have referred to admit numerous modifications and combinations. Since no firm guidelines exist for selecting matching procedures, we indicate some relevant considerations. For all methods other than the standard correlation of gray-scale imagery, the inexpensive extraction of good features is a primary concern. The cost of feature extraction must be balanced against the potential use of the features. Region features, while often costly to obtain, can be vital in later phases of image interpretation. On the other hand, isolated interest points, such as points of high curvature, can be inexpensive to obtain but serve no purpose

other than to determine registration. A second major consideration is the extent to which knowledge of the scene contents, as might be obtained from a geographic data base, should be employed in the matching procedure.



#### 4 1 Correlation-Based Techniques

Registration can be done using various correlation-based techniques. These techniques can be applied to the original gray-scale imagery or to derived pixel images such as images which contain point features or edges, or can be applied to lists of derived feature vectors.

Because of the high cost and ineffectiveness of basic correlation, other methods have been derived. This section will present the basic correlation method, a sequential correlation method and a hierarchical correlation method.

While correlation can be used to determine the approximate global translation for registration, the cost of determining the proper rotation is generally prohibitive and no provisions are made for accommodating scale change or perspective. Correlation techniques should work with images and maps, although little work has been done on this. Edge-based correlation, whether real images or abstract, give the best chance for registration.

#### 4.1.1 Basic Correlation

The correlation function  $R(u,v)$ , for a pixel image, at a point  $(u,v)$  of a  $J \times K$  window  $W$  with an  $M \times N$  image is defined as

$$R(u,v) = \frac{\sum_{j=1}^K \sum_{i=1}^J S(i,j)W(i-u,j-v)}{[\sum_{j=1}^K \sum_{i=1}^J S^2(i,j)]^{1/2} [\sum_{j=1}^K \sum_{i=1}^J W^2(i-u,j-v)]^{1/2}}$$

since  $W^2(i-u,j-v)$  is constant for a fixed  $W$ , it suffices to compute

$$R(u,v) = \frac{\sum_{j=1}^K \sum_{i=1}^J S(i,j)W(i-u,j-v)}{[\sum_{j=1}^K \sum_{i=1}^J S^2(i,j)]^{1/2}}$$

The point  $(u,v)$ , in the search image, at which  $R(u,v)$  is maximized gives the optimal location of the window. Thus our optimal match is given by the  $(u_0, v_0)$  such that

$$R(u_0, v_0) \geq R(u, v) \quad \text{for } 1 \leq u \leq (M-J+1) \\ 1 \leq v \leq (N-K+1)$$

The above pixel based matching procedure can be applied to original gray scale images, edge images and point images. These methods can be easily adapted to the case where the image is described by a feature vector, such as the vector of invariant moments of an image (see section 3.4.1). If we denote the seven invariant moments of image 1 by  $M_1, \dots, M_7$  and the invariant moments of image 2 by  $N_1, \dots, N_7$  then we define their correlation by:

$$R = \sum_{i=1}^7 M_i N_i / [\sum_{i=1}^7 M_i^2 \cdot \sum_{i=1}^7 N_i^2]$$

Invariant moment matching is in theory invariant under translation, rotation, and scale change, though in practice good results of matching radar to optical images have only been obtained for angle of up to  $45^\circ$  and scale changes of a

factor of 2, [Wong and Hall, 1978]. Invariant moment matching may be used as part of a hierarchical procedure discussed later.

Correlation-based techniques can be applied to lists of features. Of particular importance is the case where a feature such as a straight line or a triangle can be assigned to certain points in the image. Given such an assignment for two images and a transformation from one image to the other, we would like to use the features to measure the quality of the transformation as a registration. The measure will, of course, depend upon the nature of the feature. In the comparison of straight lines from two images, we can define two lines, one from each image, to be similar if the sums of the squares of the distances of their endpoints are close together. Since a triangle can be thought of as three straight lines, we can extend the previous definition to the case of triangles in a straight forward manner.

The correlation function of two line images can be defined as follows:

Let  $\alpha$  be a vector giving the parameters of a transformation mapping one image into another, e.g. translation, rotation and scale parameters.

The correlation function assigns, to each  $\alpha$ , a number  $f(\alpha)$ , measuring the quality of the transformation for registration.

To each line,  $a$ , in image one which has a line,  $b$ , in image two whose endpoints are each within a fixed value,  $e$ , of the endpoints of  $a$ , we assign the value  $D_a$  which is the sum of the distances between the endpoints of the two lines. Let  $A = \{a_1, \dots, a_N\}$  be the set of all edges in image one for which edges  $b$ , as above, exist. Define the correlation  $f(\alpha)$  by

$$f(\alpha) = \frac{1}{Ng} \left( \sum_{a_i \in A} D_{a_i} + 1 \right) \text{ if } N \neq 0$$

$$= h \quad \text{otherwise}$$

where  $h$  is a large positive constant and  $g$  is an integer parameter to be set.

The  $\alpha$  which minimizes  $f$  is said to give the best correlation.

#### 4.1.1.1 Correlating Abstract Edge Lists

An example of correlating feature vectors where the features are edges is presented. Suppose edges of Marr high curvature points, discussed in Section 3.3.2, are formed by connecting successive high curvature points. The resulting edge list does not contain real edges but, what LNK calls, abstract edges. This process was applied to the images BW-CB and IR-CB and the resulting abstract edge lists were correlated. The edges were said to correspond when the orientation of the edges were within  $\pm 30^\circ$  and the lengths of the edges were within one pixel. Part of the correlation matrix is shown in Figure 4-1. The peak occurred at displacement ( $x=-2$ ,  $y=0$ ) which is better than the straight correlation of the high curvature points. Note, the peak is much better formed than the peaks in the pixel based matching correlations.

	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
-5	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	3
-4	0	0	0	3	0	1	1	0	0	0	0	0	0	1	0	1
-3	0	0	0	1	3	2	1	0	1	1	1	0	1	0	1	1
-2	1	2	0	1	2	4	1	2	1	1	0	0	0	2	3	3
-1	0	1	4	5	5	6	4	3	2	1	0	0	1	2	2	2
0	7	8	8	10	18	<u>32</u>	15	10	3	2	3	3	5	5	3	2
+1	7	8	9	9	9	31	12	8	6	3	4	5	6	7	9	8
+2	1	2	3	5	9	17	8	8	2	3	1	2	0	0	0	1
+3	0	0	0	1	3	6	8	9	1	2	0	1	0	0	1	0
+4	0	0	0	0	2	3	2	3	3	1	2	0	0	0	0	1
+5	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0

Figure 4-1. Part of the correlation matrix of abstract edges of high curvature points.

#### 4.1.1.2 Correlating Abstract Triangle Lists

The example of the previous section was repeated but using abstract triangles formed by connecting three successive high curvature points. Figure 4-2 shows part of the resulting correlation matrix. The triangles were said to correspond if their size and orientation were similar. Again, the peak occurred at  $(x=-2, y=0)$ . These two experiments show that correlation using abstract features can work much better than the basic pixel-based correlation.

X														
	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-4	0	0	0	2	0	1	1	0	0	0	0	0	0	0
-3	0	0	0	1	2	2	1	0	1	1	0	0	1	0
-2	0	2	0	1	1	4	1	2	1	0	0	0	0	1
-1	0	0	4	5	4	6	3	3	0	0	0	0	1	2
0	4	5	6	8	14	28	11	6	2	1	0	1	1	3
1	6	7	7	9	7	21	12	7	5	2	3	4	5	6
2	0	1	2	2	7	13	7	7	1	2	1	1	0	0
3	0	0	0	0	3	4	6	8	1	2	0	0	0	0
4	0	0	0	0	1	2	2	2	2	1	2	0	0	0
Y														

Figure 4-2. Part of the correlation matrix using abstract triangles of high curvature points.

#### 4.1.2 Sequential Matching Methods

One method for reducing the amount of computation in performing matching is to do the matching sequentially [Barnea & Silverman 1972]. Suppose there is an  $L \times L$  array of pixels,  $W$ , called a window and an  $M \times M$  array of pixels,  $S$ , called a search area with  $M \leq L$ . The aim of the procedure is to find that  $M \times M$  subarray of  $S$  which correlates best with  $W$ . Let  $S_M^{i,j}$  denote the  $M \times M$  subimage of  $S$  with the reference point  $(i,j)$  at the upper left corner. Define coordinates in the array  $S_M^{i,j}$  by

$$\begin{aligned} S_M^{i,j}(n,m) &\equiv S(i+n-1, j+m-1) \\ &\text{for } 1 \leq n, m \leq M \\ &1 \leq i, j \leq L-M+1. \end{aligned}$$

The window  $W$  is compared with its subwindow  $S_M^{i,j}$  by comparing a subset of its corresponding pixel pairs, defined by  $\langle S_M^{i,j}(n,m), W(n,m) \rangle$ . The possible number of comparisons required is  $M^2(L-M+1)^2$ , since there are  $(L-M+1)^2$  windows each of size  $M^2$ . The sequential matching method requires examining all  $(L-M+1)^2$  windows but not, in general, all of the  $M^2$  pixel pairs. A measure of dissimilarity between pixel pairs is defined so that matching of windows can be accomplished by accumulating the dissimilarity of pixel pairs until a threshold is reached. In order to do this, the pixel pairs must be ordered in some way.

One procedure for ordering the  $M^2$  pixel pairs is by randomly generating a permutation of the numbers  $1, \dots, M^2$  and examining the pairs in the order specified by the permutation. Then two measures of dissimilarity, or error, between the components of the pixel pair  $S_M^{i,j}(p,q)$  and  $W(p,q)$  can be defined. They are  $\epsilon'$ , the unnormalized measure, and  $\epsilon$ , the normalized measure, given by

$$\begin{aligned} \epsilon'(i,j,p,q) &\equiv |S_M^{i,j}(p,q) - W(p,q)|, \\ \epsilon(i,j,p,q) &\equiv |S_M^{i,j}(p,q) - \hat{S}(i,j) - W(p,q) + \bar{W}|, \end{aligned}$$



where

$$\bar{W} \equiv \frac{1}{M^2} \sum_{n=1}^M \sum_{m=1}^M W(n,m),$$

and

$$\hat{S}(i,j) \equiv \frac{1}{M^2} \sum_{n=1}^M \sum_{m=1}^M S_M^{i,j}(n,m).$$

The errors  $\epsilon$  or  $\epsilon'$  are accumulated by comparing the pixel pairs in the order specified by the permutation, until a given threshold is reached. The number of pixel pairs, of  $S_M^{i,j}$  and  $W$ , compared before the threshold is surpassed is stored in an array  $F$  at the location  $F(i,j)$ . The points  $(i,j)$ , for which  $F(i,j)$  is large correspond to potential window matches. The potential window matches can then be used by other, more exhaustive, matching procedures as initial guesses.

Variations of the sequential method have been described [Barnea & Silverman, 1972; Ramapriyan, 1976]. For example, the threshold may be given as a function of the rate of change of accumulated error with respect to number of point pairs examined so far. In addition, other methods for ordering the pairs are possible.

As with basic correlation, sequential correlation is designed to determine a translation for registration. Rotational correction is still expensive and scale changes are not considered. Hall describes sequential correlation of edge images which could also be applied to the registration of an image to a map. This edge image correlation may be applied to images from dissimilar sensors.

#### 4.1.3 Hierarchical Scene Matching

Hierarchical scene matching [Wong & Hall 1978] is a technique for registering one image with another, using a sequence of images derived from the original images, but varying in size and resolution. Each successive set of images are smaller in size and have lower resolution. Let level  $k=0$  correspond to the original image which is assumed to be square. The  $(k+1)^{st}$  set of images is found, from the  $k^{th}$  set, by partitioning each  $k$ -level image into four equal size square windows, applying a low-pass filter to each window and then sampling at  $1/2$  the sample rate of the previous level. This gives a set of smaller windows at a lower resolution.

At some point, this process is stopped and the highest level subimages of the two original images can be compared. Many registration methods could be used for the comparisons, such as the previously discussed sequential method.

The hierarchical technique need not be applied to the original gray-scale image. Instead binary edge images could be extracted and the hierarchical process could be applied to them, for example. Or, each subimage could be represented by a set of the seven invariant moments presented in section 3.4.1 and these could be correlated.

As in Section 4.1.2, hierarchical correlation is designed to determine a registration translation. Due to the computational savings offered by this method over ordinary correlation, it may be possible to take into account rotation and scale changes. Hall [1979] describes an application of this method to matching optical to radar images. In this work he performs the hierarchical matching on edge images formed from the original images. This method could be adapted to the matching of an image to a map.

#### 4.2. Basic L.N.K. Registration Technique

This section presents an overview of the basic LNK registration procedure. A more detailed description can be found in Appendix A. The goal of registration is to use automatically extracted features from two images (or an image and a map) to find the best global transformation that maps one image to the other.

The LNK procedure is able to achieve this even when many local mismatches of features occur. The three basic steps of the method are

- 1) Primitive features, such as line segments, edge segments, intersections, or high curvature points are automatically extracted. These features may be parameterized, such as by position, by length, or by orientation.
- 2) Assume all features of one type can correspond to one another. For example, an intersection of three lines in one image can correspond to any three-line intersection in the second image. However, only one of these correspondences is the true one. Now, for each of the possible correspondences, find the translation and rotation that maps one feature to the other. Let that translation and rotation transformation be represented by the triple  $(\Delta\theta, \Delta x, \Delta y)$ . Place a unit of weight in the bin in the three dimensional histogram that represents  $(\Delta\theta, \Delta x, \Delta y)$ . This process is repeated until all possible feature correspondences have been transformed.
- 3) Locate any prominent clusters of  $(\Delta\theta, \Delta x, \Delta y)$  in the histogram. Each cluster represents a set of features in one image that could be mapped to corresponding features in the other image by one particular  $(\Delta\theta, \Delta x, \Delta y)$ . The best global transformation is defined by the  $(\Delta\theta, \Delta x, \Delta y)$  of the largest cluster. This transformation provides the largest number of local correspondences.

This method works because the correct transformation will show up as a large cluster, while the wrong transformations will tend to be distributed randomly throughout the histogram. The procedure may also be performed iteratively by making the bin sizes smaller and smaller.

As an example, suppose an image can be represented by the four directed edge segments shown in Figure 4-3(a). The direction of the arrow is determined by making the region on the right of the edge segment the darker one.

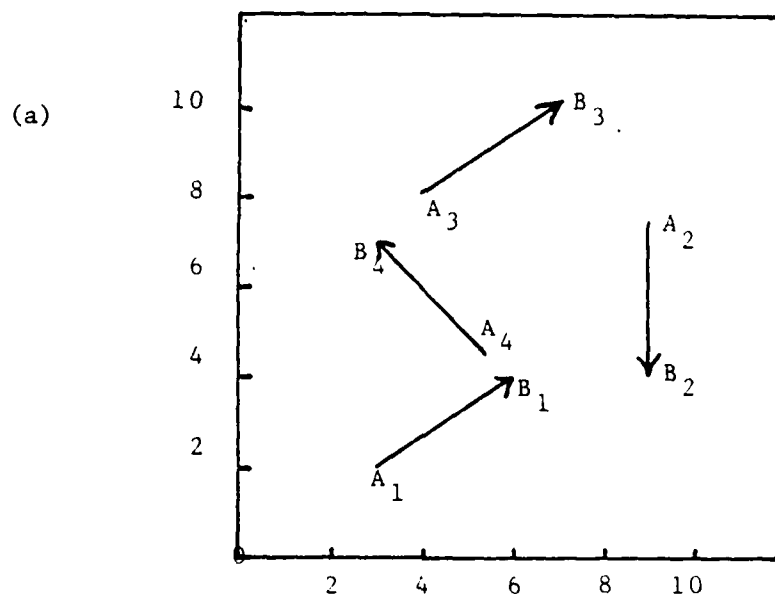
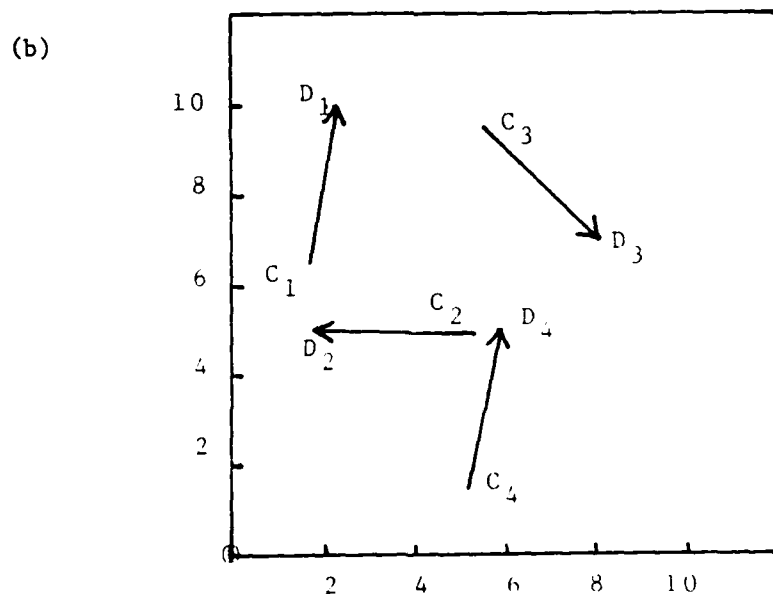


Figure 4-3. Example of basic LNK registration technique. Image edge elements in (a) need to be rotated  $45^\circ$  and then translated (4.5, -2.0) to be transformed into corresponding map edge elements in (b).



Suppose the map or second image contains the edge segments in Figure 4-3(b). There are 16 possible ways that the edge segments in (a) can be paired with the edge segments in (b). Four of the 16 pairings yield a consistent interpretation--rotate by  $\Theta=4.5^\circ$  and translate by (4.5, 2.0). The triple ( $\Theta=45^\circ$ ,  $\Delta x=4.5$ ,  $\Delta y=2.0$ ) forms a cluster in the three-dimensional histogram while the triples from the incorrect pairings are sparsely distributed. Table 4-1 shows the 16 possible transformations. The correct transformations are indicated by an '\*'. In actual cases, there will be many more than 4 primitive features and not all pairings will be possible (i.e. due to size, shape, or type differences) so the cluster of correct transformations should be even more prominent.

As stated in step 1 of the procedure, many features could be used to perform the transformation. Not all features are equally desirable. For example, the lengths of edge segments or line segments usually can not be accurately determined. Small perspective changes can alter the curvature of high curvature points considerably. LNK feels that intersections would produce more accurate transformations as they can be determined more accurately. LNK has performed enough experiments to show that even if up to 90% of the detected features in the image do not match with any features in the map, the correct transformation will still form enough of a cluster to be detected.

Table 4-1

Transformations of all 16 possible combinations of image and map vectors from Figure 4-3. A cluster (indicated by '\*') was formed at ( $\theta = 0.79$ ,  $\Delta x = 4.5$ ,  $\Delta y = -2.0$ ).

i	j	A	B	C	D		θ	xs	ys
1	1	3.0,2.0	6.0,4.0	1.7,6.4	2.3,10.0		0.82	1.1	2.8
1	2	3.0,2.0	6.0,4.0	5.3,5.0	1.8,5.0		2.55	8.9	5.0
1	3	3.0,2.0	6.0,4.0	5.5,9.5	8.0,7.0		4.91	3.0	12.1
1	4	3.0,2.0	6.0,4.0	5.1,1.5	5.8,5.0		0.79	4.4	-2.0
2	1	9.0,7.5	9.0,4.0	1.7,6.4	2.3,10.0		2.98	11.8	12.3
2	2	9.0,7.5	9.0,4.0	5.3,5.0	1.8,5.0		4.71	-2.2	14.0
2	3	9.0,7.5	9.0,4.0	5.5,9.5	8.0,7.0		0.79	4.4	-2.2
2	4	9.0,7.5	9.0,4.0	5.1,1.5	5.8,5.0		2.94	15.4	7.1
3	1	4.0,8.0	7.0,10.0	1.7,6.4	2.3,10.0		0.82	4.8	-2.0
3	2	4.0,8.0	7.0,10.0	5.3,5.0	1.8,5.0		2.55	13.1	9.4
3	3	4.0,8.0	7.0,10.0	5.5,9.5	8.0,7.0		4.91	-3.1	11.9
3	4	4.0,8.0	7.0,10.0	5.1,1.5	5.8,5.0		0.79	7.9	-7.0
4	1	5.5,4.5	3.0,7.0	1.7,6.4	2.3,10.0		5.33	-5.2	3.3
4	2	5.5,4.5	3.0,7.0	5.3,5.0	1.8,5.0		0.79	4.6	-2.1
4	3	5.5,4.5	3.0,7.0	5.5,9.5	8.0,7.0		3.14	11.0	14.0
4	4	5.5,4.5	3.0,7.0	5.1,1.5	5.8,5.0		5.30	-1.7	3.6

#### 4.2.1 Example

The basic LNK registration technique was applied to the image in Figure 4-4. The straight line Hough detector was applied to this image and the result is shown in Figure 4-5. Another view of the naval base, shown in Figure 4-6, was used to create a map of edge segments, shown in Figure 4-7. Three ground control points were used in each image to establish the approximate registration transformation, ( $\theta=328^\circ$ ,  $\Delta x=-141$ ,  $\Delta y=9$ ).

A hierarchical clustering technique was used to determine the results. In this method, clustering is first done on  $\theta$  and then in  $(\Delta x, \Delta y)$  space given a fixed  $\theta$ . The result is shown in Table 4-2. The strongest transformation, ( $\theta=330^\circ$ ,  $\Delta x=-142$ ,  $\Delta y=8$ ) aligned 19 out of 200 image edge segments with 12 out of 22 map edge segments. The strongest contender ( $\theta=237^\circ$ ,  $\Delta x=453$ ,  $\Delta y=-433$ ) aligned only 6 image edge segments with 2 map edge segments. The results would probably be enhanced if a more comprehensive map had been constructed.

Table 4-2

Summary of Results Using the  
Basic LNK Registration Technique

index	$\theta$ cluster center/radius/strength	index	$(\Delta x, \Delta y)$ cluster center/radius/strength
cluster 1	$330^\circ/3^\circ/176$	cluster 11	$(-71, -215)/10/ 2.35$
		cluster 12	$(-142, 8) /10/12.85$
		cluster 13	$(195, 135) /10/ 5.04$
cluster 2	$237^\circ/3^\circ/152$	cluster 21	$(453, -433)/20/ 5.14$
		cluster 22	$(345, 78) /10/ 3.48$
		cluster 23	$(276, 146) /10/ 3.86$
cluster 3	$61^\circ/3^\circ/139$	cluster 31	$(-22, -406)/20/ 1.28$
		— *	
		— *	

\* indicates no viable alternative cluster



Figure 4-4. UNB1 Test image: Naval base photo containing network of curving roads.



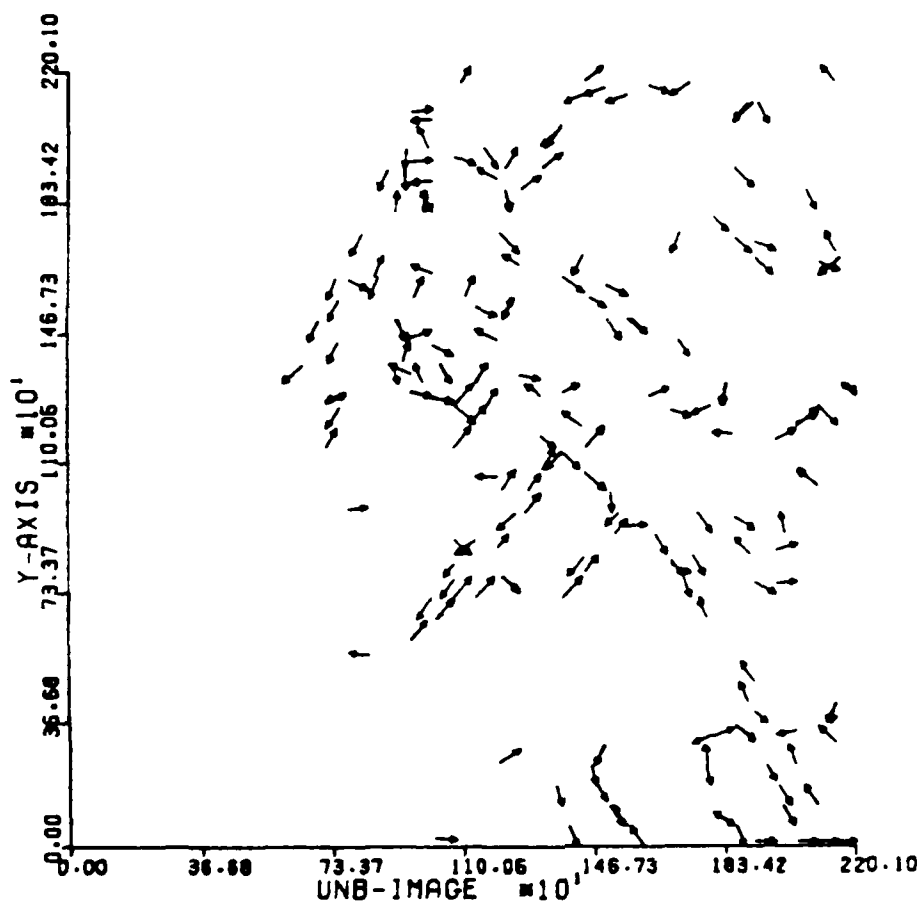


Figure 4-5. Hough detections from UNB1 Figure 4-4 image with  $2^\circ$  directional resolution.



Figure 4-6. UNB2: Second photo of naval base shown in Figure 4-4.

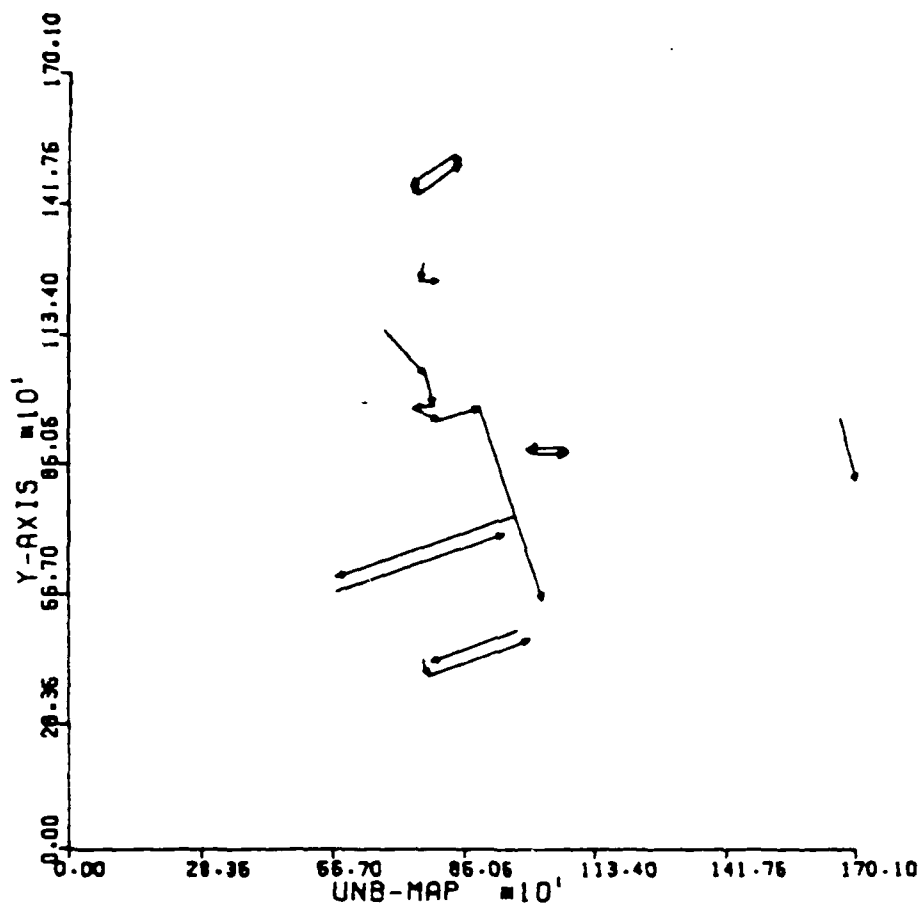


Figure 4-7. UNB edge element map made by human from UNB2, Figure 4-6.

#### 4.2.2 Full RST Transformation

The position and orientation of primitive features such as intersections, high curvature points, or lines can accurately be determined, but it is difficult to determine their sizes, such as length. The basic LNK registration technique uses the position and orientation of the primitive features to find the rotation and translation necessary to register two images or to register an image to a map.

If the size could be included, then it would be possible to calculate the four parameter transformation of rotation, translation and scaling. This section presents a method for extending the procedure to account for scaling. In order to accomplish this, abstract vectors or edges whose size can accurately be determined are introduced.

To achieve scaling, abstract vectors or edges can be formed by spanning pairs of point structures. For example, abstract edges can be formed by connecting pairs of high curvature points. Abstract vectors could be formed by using an intersection point as the vector tail and a high curvature point as the vector head. There are many ways of forming the abstract edges or vectors.

The registration procedure is similar to the basic one. Instead of the triples  $(\theta, \Delta x, \Delta y)$ , there are four parameters  $(\theta, \Delta x, \Delta y, \Delta s)$ , where  $\Delta s$  is the scaling parameter.

The three registration steps are now: (Appendix A provides a detailed description)

- 1) Primitive point features (intersections, high curvature points, etc) are automatically extracted. The abstract vectors are created by pairing the primitive point features. Not all possible pairs need be formed as that would result in a great deal of computation.
- 2) Assume all features of one type can correspond to one another. That is, a vector from an intersection of three lines to an intersection of four lines in the first image can correspond to any 3-intersection to 4-intersection vector extracted from

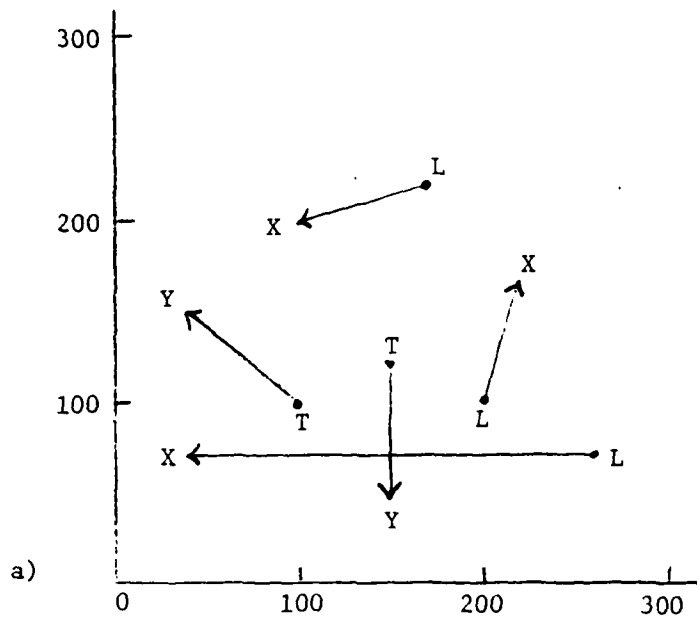
the map (or second image). For each possible correspondence, find the 4-parameter transformation ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) that maps one vector to the other. Place a unit of weight in the bin in the four dimensional histogram that represents the ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) found.

- 3) Locate the best cluster in the histogram. The ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) of that cluster is the best global transformation as it provided the largest number of local correspondences.

As an example of this method, suppose an image and map are represented in terms of vectors  $v_j$  connecting intersection points. The intersection points are of four types; 'L', 'T', 'X', or 'Y'. The rules for pairing points to form vectors may be arbitrary; for example, 'T' points are joined with 'L' points. The purpose of such rules is to control the combinatorics. In Figure 4-8, five vectors represent the map and four vectors represent the image.

Given any map vector  $v_j$  all possible matching image vectors  $v_i$  are considered. Each possible pairing ( $v_j, v_i$ ) results in an RS&T transformation mapping image vector  $v_i$  onto map vector  $v_j$ . The transformation is specified by a quad of parameters ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) where  $\Delta\theta$  is the angle of rotation,  $\Delta s$  is the scale factor, and  $\Delta x$  and  $\Delta y$  are the x and y translations respectively. The pair ( $v_j, v_i$ ) is discarded without producing a quad if the tips or tails of the vectors  $v_j$  and  $v_i$  disagree in type. In the example of Figure 4-8 there are  $5 \times 4 = 20$  pairs ( $v_j, v_i$ ) initially possible and of these only 10 agree in type of tip and tail.

The mathematical development for forming a quad ( $\Delta\theta$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta s$ ) as a function of  $v_j$  and  $v_i$  is given in Figure 4-9. Table 4-3 shows computer output for the example shown in Figure 4-8. There are 10 quads produced and a cluster of size 3 is evident in the neighborhood of the best registration transformation ( $\Delta\theta=5.10$ ,  $\Delta x=-75$ ,  $\Delta y=88$ ,  $\Delta s=0.5$ ). In real world cases there would be hundreds of quads overall and a few dozen in a cluster.



Type Code

L 1

T 2

X 3

Y 4

Map Vectors

(L,170,220) - (X,100,200)

(T,100,100) - (Y,40,150)

(L,200,100) - (X,220,170)

(L,260,70) - (X,40,70)

(T,150,125) - (Y,150,50)

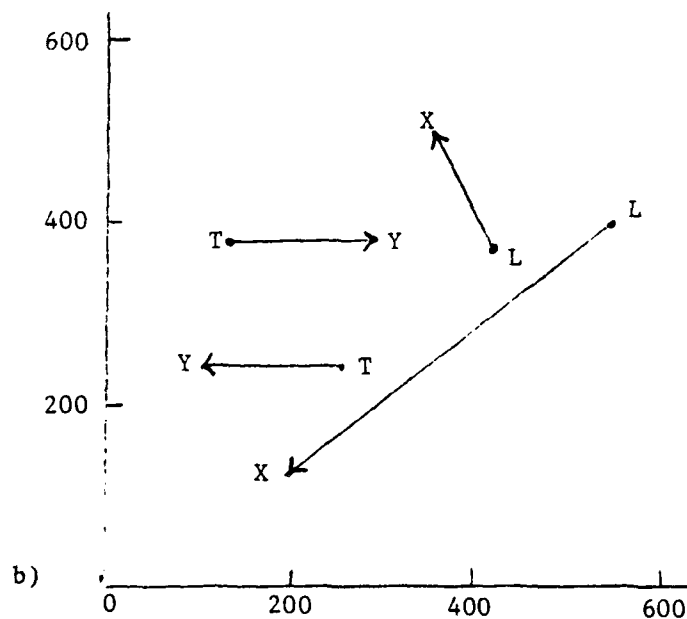


Image Vectors

(L,545,400) - (X,200,120)

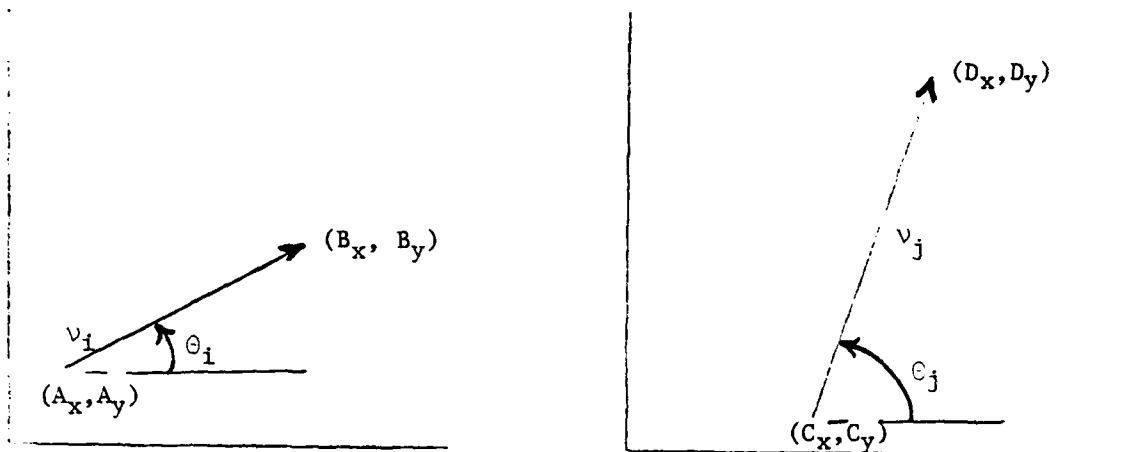
(T,260,240) - (Y,100,245)

(T,140,380) - (Y,300,380)

(L,420,370) - (X,360,500)

Figure 4-8. a) Example set of map vectors, and

b) set of vectors representing an image to be registered to the map by RS&T transformation.



Assuming that vector  $v_i$  corresponds to vector  $v_j$  transformation parameters  $(\Delta\theta, \Delta x, \Delta y, \Delta s)$  are gotten as

$$\Delta\theta = \theta_j - \theta_i$$

$$\Delta s = \text{length of } v_j / \text{length of } v_i$$

$$\Delta x = \Delta s A_y \sin\Delta\theta - \Delta s A_x \cos\Delta\theta + C_x$$

$$\Delta y = -\Delta s A_x \sin\Delta\theta - \Delta s A_y \cos\Delta\theta + C_y$$

The resulting registration transformation in homogeneous coordinates is

$$[u, v, 1] = [x, y, 1] \begin{bmatrix} \Delta s \cos\Delta\theta & \Delta s \sin\Delta\theta & 0 \\ -\Delta s \sin\Delta\theta & \Delta s \cos\Delta\theta & 0 \\ \Delta x & \Delta y & 1 \end{bmatrix}$$

where  $(x, y)$  is an image point and  $(u, v)$  is the corresponding map point.

Figure 4-9. Mathematical derivation of RS&T transformation parameters from a pair of vectors  $v_i$  and  $v_j$  assumed to be corresponding.

Table 4-3

Ten possible pairings of image and map vectors produce a cluster in RS&T parameter space near ( $\Delta\theta=5.6, \Delta x=-75, \Delta y=88, \Delta s=0.5$ )

MAP VECTOR						IMAGE VECTOR			TRANSFORMATION			
VEC #	TAIL TYP(X,Y)	HEAD TYP(X,Y)	VEC #	TAIL TYP(X,Y)	HEAD TYP(X,Y)	THETA	SCALE	DELX	DELY			
1	1( 170, 220)	3( 100, 200)	6	1( 545, 400)	3( 200, 120)	.588+01	.164+00	.621+02	.195+03			
1	1( 170, 220)	3( 100, 200)	9	1( 420, 370)	3( 360, 500)	.142+01	.508+00	.323+03	-.199+02			
2	2( 100, 100)	4( 40, 150)	7	2( 260, 240)	4( 100, 245)	.562+01	.488+00	-.721+02	.859+02 *			
2	2( 100, 100)	4( 40, 150)	8	2( 140, 380)	4( 300, 380)	.245+01	.488+00	.271+03	.199+03			
3	1( 200, 100)	3( 220, 170)	6	1( 545, 400)	3( 200, 120)	.375+01	.164+00	.236+03	.205+03			
3	1( 200, 100)	3( 220, 170)	9	1( 420, 370)	3( 360, 500)	.557+01	.508+00	-.846+02	.967+02 *			
4	1( 260, 70)	3( 40, 70)	6	1( 545, 400)	3( 200, 120)	.560+01	.495+00	-.743+02	.863+02 *			
4	1( 260, 70)	3( 40, 70)	9	1( 420, 370)	3( 360, 500)	.114+01	.154+01	.506+03	-.754+03			
5	2( 150, 125)	4( 150, 50)	7	2( 260, 240)	4( 100, 245)	.160+01	.469+00	.266+03	.676+01			
5	2( 150, 125)	4( 150, 50)	8	2( 140, 380)	4( 300, 380)	.471+01	.469+00	-.281+02	.191+03			



#### 4.2.2.1 Example with Scale

The following experiment demonstrates the utility of the proposed technique for regions where cultural activity creates features such as straight edges or networks of lineals. The results reported are typical of many similar experiments.

Point features were identified by eye on an aerial image from the mid-western U.S.A. using two different measuring devices and two different orientations. Figure 4-10 shows sample imagery while Table 4-4 contains the coordinates of the selected feature points. Points are labeled according to the type of road intersection - 'L', 'T', 'X', or 'Y' - which they represent or are labeled 'A' indicating an arbitrary point on the road.

50 vectors were chosen to model the map of Figure 4-10 while 64 vectors represented the image. Tables 4-5 and 4-6 show some of the resulting quads formed by matching vectors from the map with vectors from the image. Note that stage coordinates have been divided by 10 for format convenience. Table 4-5 shows 10 vectors in stage coordinates which have the same type of tip and tail as the vector (4365,2747) - (2498,3641) in the photo model. Only the first of these matches is correct and hence only the first quad (5.39, -1020,4040. 5.01) contributes to the ultimate cluster. Of the  $50 \times 64 = 3200$  pairs  $(v_j, v_i)$  possible only 790 produce quads in cluster space after the check on tip and tail type. Table 4-6 shows that 32 of the 790 quads form a cluster near the parameter set  $(\Delta\theta=5.38, \Delta x=-1000., \Delta y=4000., \Delta s=5.0)$ . Using Table 4-6 the reader can verify that 30 of these 32 quads represent correct vector matches. For instance, the first quad represents the matching of vectors from point #2 to point #1 in the two different coordinate systems. The asterisks mark the two incorrect matches which in fact are outliers in the set of 32 quads. A

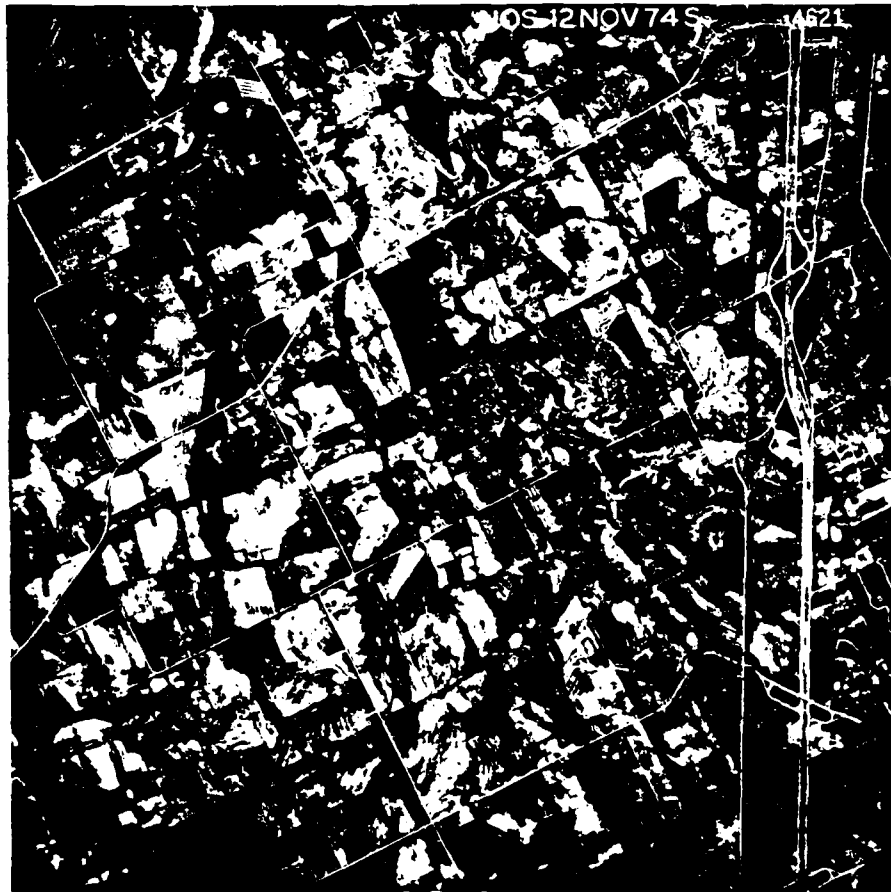


Figure 4-10. Aerial photo over midwestern U.S.A. referred to as "4621". Coordinates of labeled points are given in Table 4-4.

Table 4-4

Digitization of points on image 4621 using Talos digitizer  
(0.001 inch resolution) and scanning stage (0.0005 inch  
resolution).

IMAGE PT #	TALOS *		STAGE *		DESCRIPTION **
	(photo)		(transparency)		
	X	Y	X	Y	
1	2498	3641	5000	5000	X
2	4365	2747	8722	6803	T
3	5583	2191	11120	8052	X
4	3044	4744	3913	7203	T
5	3342	5357	3314	8405	T
6	3611	5908	2814	9511	T
7	1980	6678	-	-	T
8	3975	5750	3508	9946	T
9	5516	5073	6563	11448	T
10	7130	3514	10964	12108	Y
11	7430	4091	10367	13304	L
12	7723	4700	9866	14600	L
13	8642	4258	-	-	X
14	6952	3223	11209	11461	A
15	6942	3613	10565	12000	L
16	7233	4181	10010	13101	T
17	4861	7575	1710	13600	X
18	5980	7667	-	-	Y
19	6444	7480	-	-	T
20	9471	6190	-	-	T
21	5013	7886	1399	14295	A
22	4480	7729	-	-	T
23	7346	9632	-	-	A
24	9282	5763	-	-	A
25	9700	6104	-	-	Y
26	5279	7999	1513	14804	A
27	4010	1991	9388	5480	A
28	5480	2552	-	-	L
29	4808	2527	-	-	A

\* The photo in Figure 4-10 was rotated roughly 50 degrees clockwise when mounted on the digitizer. The corresponding transparency was not rotated when mounted on the stage.

\*\* L, X, T, and Y indicate type of intersection, while A indicates an arbitrary point on a straight road segment.

Table 4-5.

First 10 quads in set of 790 quads produced by comparing 50 map vectors with 64 image vectors. (Stage coordinates of Table 4-4 divided by 10 for format convenience).

<u>MAP VECTOR</u>			<u>IMAGE VECTOR</u>			<u>TRANSFORMATION</u>			
VEC #	TAIL TYP(X,Y)	HEAD TYP(X,Y)	VEC #	TAIL TYP(X,Y)	HEAD TYP(X,Y)	THETA	SCALE	DELX	DELY
1	3(4365,2747)	4(2498,3641)	51	3( 872, 680)	4( 500, 500)	.539+01	.501+01	-.102+04	.404+04
1	3(4365,2747)	4(2498,3641)	52	3( 391, 720)	4( 500, 500)	.381+01	.843+01	.322+04	.956+04
1	3(4365,2747)	4(2498,3641)	53	3( 331, 840)	4( 500, 500)	.380+01	.545+01	.297+04	.747+04
1	3(4365,2747)	4(2498,3641)	54	3( 281, 951)	4( 500, 500)	.381+01	.413+01	.283+04	.654+04
1	3(4365,2747)	4(2498,3641)	55	3( 350, 994)	4( 500, 500)	.397+01	.401+01	.237+04	.647+04
1	3(4365,2747)	4(2498,3641)	56	3( 656,1144)	4( 500, 500)	.450+01	.312+01	.129+04	.549+04
1	3(4365,2747)	4(2498,3641)	61	3(1001,1310)	4( 500, 500)	.482+01	.217+01	.130+04	.461+04
1	3(4365,2747)	4(2498,3641)	62	3( 872, 680)	4(1112, 805)	.221+01	.765+01	.125+05	.536+03
1	3(4365,2747)	4(2498,3641)	67	3( 872, 680)	4( 171,1360)	.324+00	.212+01	.307+04	.793+03
1	3(4365,2747)	4(2498,3641)	68	3( 391, 720)	4(1112, 805)	.258+01	.285+01	.640+04	.389+04

Table 4-6

32 quads contained in the bin  $\Theta \in [5.0, 5.7]$ ,  $S \in [4.8, 5.2]$ ,  
 $\Delta x \in [-1100, -900]$ , and  $\Delta y \in [3000, 5000]$ .

<u>MAP VECTOR</u>			<u>IMAGE VECTOR</u>			<u>TRANSFORMATION</u>			
VEC #	TAIL TYP(X,Y)	HEAD TYP(X,Y)	VEC #	TAIL TYP(X,Y)	HEAD TYP(X,Y)	THETA	SCALE	DELX	DELY
1	3(4365,2747)	4(2498,3641)	51	3( 872, 680)	4( 500, 500)	.539+01	.501+01-	.102+04	.404+04
2	3(3044,4744)	4(2498,3641)	52	3( 391, 720)	4( 500, 500)	.536+01	.501+01-	.101+04	.412+04
3	3(3342,5357)	4(2498,3641)	53	3( 331, 840)	4( 500, 500)	.536+01	.504+01-	.103+04	.411+04
4	3(3611,5908)	4(2498,3641)	54	3( 281, 951)	4( 500, 500)	.537+01	.504+01-	.104+04	.408+04
4	3(3611,5908)	4(2498,3641)	55	3( 350, 994)	4( 500, 500)	.553+01	.489+01-	.959+03	.352+04
6	3(3975,5750)	4(2498,3641)	54	3( 281, 951)	4( 500, 500)	.522+01	.514+01-	.994+03	.464+04
6	3(3975,5750)	4(2498,3641)	55	3( 350, 994)	4( 500, 500)	.538+01	.499+01-	.100+04	.406+04
7	3(5516,5073)	4(2498,3641)	56	3( 656,1144)	4( 500, 500)	.539+01	.504+01-	.105+04	.401+04
8	4(2498,3641)	5(7130,3514)	57	4( 500, 500)	5(1096,1210)	.538+01	.500+01-	.101+04	.404+04
9	1(7430,4091)	4(2498,3641)	58	1(1036,1330)	4( 500, 500)	.538+01	.501+01-	.102+04	.407+04
10	1(7723,4700)	4(2498,3641)	59	1( 986,1460)	4( 500, 500)	.538+01	.495+01-	.982+03	.405+04
11	1(6942,3613)	4(2498,3641)	60	1(1056,1200)	4( 500, 500)	.538+01	.497+01-	.992+03	.406+04
12	3(7233,4181)	4(2498,3641)	61	3(1001,1310)	4( 500, 500)	.538+01	.500+01-	.102+04	.406+04
19	3(4365,2747)	4(5583,2191)	62	3( 872, 680)	4(1112, 805)	.537+01	.495+01-	.941+03	.408+04
20	3(4365,2747)	5(7130,3514)	63	3( 872, 680)	5(1096,1210)	.538+01	.499+01-	.994+03	.405+04
21	1(7430,4091)	3(4365,2747)	64	1(1036,1330)	3( 872, 680)	.537+01	.499+01-	.987+03	.410+04
22	1(7723,4700)	3(4365,2747)	65	1( 986,1460)	3( 872, 680)	.538+01	.493+01-	.933+03	.402+04
24	1(6942,3613)	3(4365,2747)	66	1(1056,1200)	3( 872, 680)	.538+01	.493+01-	.923+03	.406+04
25	3(4365,2747)	4(4861,7575)	67	3( 872, 680)	4( 171,1360)	.538+01	.497+01-	.972+03	.406+04
29	3(3044,4744)	4(5583,2191)	68	3( 391, 720)	4(1112, 805)	.538+01	.496+01-	.962+03	.407+04
30	3(3342,5357)	4(5583,2191)	69	3( 331, 840)	4(1112, 805)	.537+01	.496+01-	.956+03	.410+04
31	3(3611,5908)	4(5583,2191)	70	3( 281, 951)	4(1112, 805)	.537+01	.499+01-	.992+03	.410+04
33	3(3975,5750)	4(5583,2191)	71	3( 350, 994)	4(1112, 805)	.538+01	.497+01-	.986+03	.406+04
35	4(5583,2191)	5(7130,3514)	73	4(1112, 805)	5(1096,1210)	.538+01	.502+01-	.105+04	.407+04
36	1(7430,4091)	4(5583,2191)	74	1(1036,1330)	4(1112, 805)	.537+01	.500+01-	.991+03	.414+04
37	1(7723,4700)	4(5583,2191)	75	1( 986,1460)	4(1112, 805)	.539+01	.494+01-	.959+03	.400+04
38	1(6942,3613)	4(5583,2191)	76	1(1056,1200)	4(1112, 805)	.538+01	.493+01-	.927+03	.404+04
39	3(7233,4181)	4(5583,2191)	77	3(1001,1310)	4(1112, 805)	.537+01	.500+01-	.101+04	.410+04
46	3(3044,4744)	5(7130,3514)	78	3( 391, 720)	5(1096,1210)	.538+01	.497+01-	.967+03	.404+04
47	1(7430,4091)	3(3044,4744)	79	1(1036,1330)	3( 391, 720)	.538+01	.499+01-	.991+03	.406+04
48	1(7723,4700)	3(3044,4744)	80	1( 986,1460)	3( 391, 720)	.538+01	.493+01-	.935+03	.406+04
50	1(6942,3613)	3(3044,4744)	81	1(1056,1200)	3( 391, 720)	.538+01	.495+01-	.955+03	.408+04

4-D clustering routine would need to be implemented to find the significant clusters.

#### 4.2.3 Extending LNK Registration to 3 Dimensions

So far the LNK registration procedures have ignored the 3D aspects of the objects. This section presents a method to extend the 2D registration to 3D under certain constraints. The 2D procedure found RS&T transformations mapping map structure onto image structure. Experiments showed that the RS&T assumption was feasible in cases where variation in the 3rd dimension was relatively small. Good approximate RS&T registration transformations were obtained automatically for photo/map pairs even when there was some relief in the terrain. There are many cases where RS&T transformations are inadequate, such as in low altitude aerial imaging and in acquisition of solid objects by a robot vision system. In these cases projective transformations must be used.

In the general case 6 parameters are necessary to specify imaging in a 3D world [Duda and Hart 1973]. In this section, we consider a constrained imaging system with only 3 free parameters as shown in Figure 4-11. A front image plane is used with reference system origin ( $x=0, y=0, z=0$ ) at the image center. The camera has known focal length  $f$  and looks vertically down at a scene with base distance  $y_0$  from the image plane. There are only 3 unknown parameters to discover; the angle  $\theta$  at which the object lies on the base plane and the amount of translation ( $x_0, z_0$ ) of the object origin from the point where the camera axis pierces the plane.

There is some justification for this assumption in the aerial imaging case. First,  $f$  is usually known. Second, it is possible to get a good approximation for altitude  $y_0$  and to achieve a nearly vertical camera axis. These approximations would perhaps be good enough to correctly detect an approximate 3 parameter view which could be used as an initial approximation

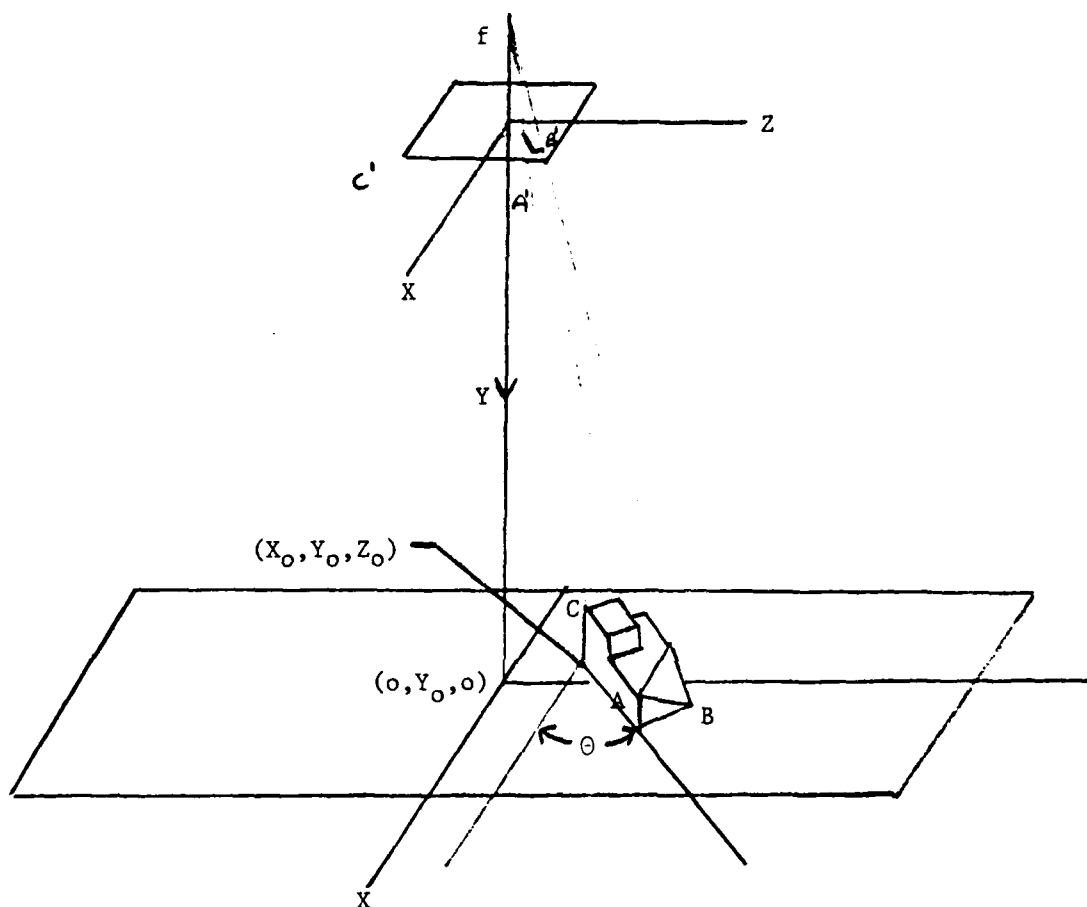


Figure 4-11. Three dimensional object viewed by camera with known attitude. Knowns are  $f$  and  $Y_o$  and a model of the object. Unknowns are the object orientation parameters  $\theta$ ,  $X_o$ , and  $Z_o$ .



to a full 6 parameter view.

We therefore proceed as follows. A 3D terrain model is specified in local coordinates as in Figure 4-12. The model would contain the location and description of all significant features such as edges, corners, intersections, water bodies, etc.

The acquisition problem is defined as discovering (computing) the orientation parameters  $(\theta, x_o, z_o)$  from the image structure and the known camera parameters  $f$  and  $y_o$ . A few definitions are appropriate before proceeding.

camera parameters - parameters that fix the imaging system over the base plane, i.e.  $f$  and  $y_o$ , and define the global coordinate system.

orientation parameters - parameters that fix the object (or object model) in the global coordinate system which are  $\theta$ ,  $x_o$ ,  $z_o$ .

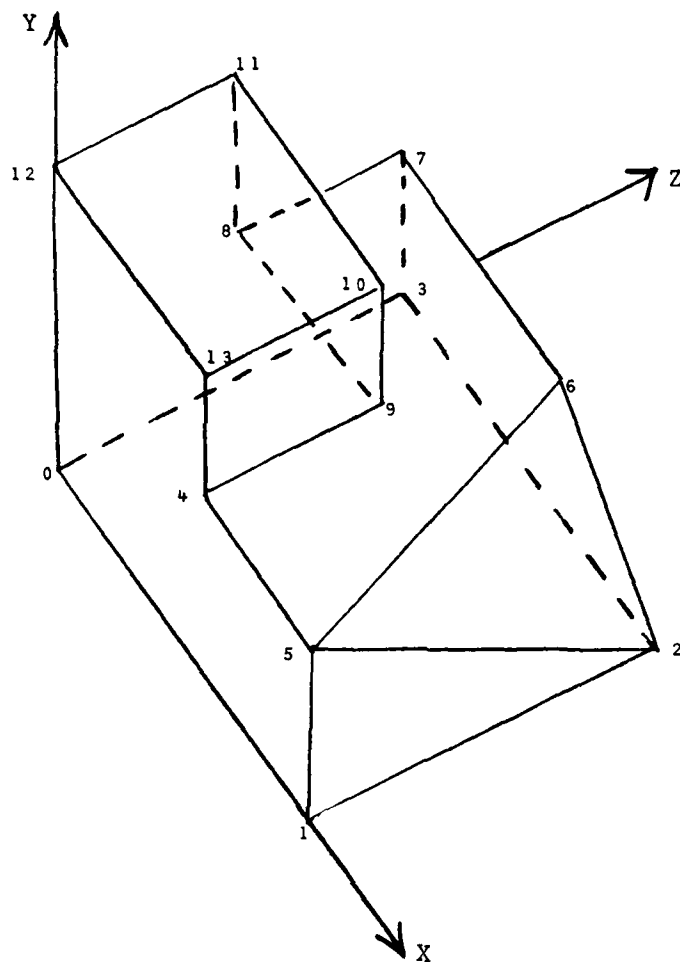
viewing parameters - the combined camera and orientation parameters  $f, y_o, \theta, x_o, z_o$  which allow a specific image to be created (from the terrain model).

Here we assume that we know the attitude of the camera  $f, y_o$  and the orientation  $\theta, x_o, z_o$  of the object. We develop computational formulas for image point  $(x_I, y_I, z_I)$  corresponding to point  $(x_m, y_m, z_m)$  in the map.

Let  $(x, y, z)$  be the global coordinates of point  $(x_m, y_m, z_m)$  under map orientation  $(\theta, x_o, z_o)$ . Then we have

$$\begin{aligned} (1) \quad x &= x_m \cos \theta - z_m \sin \theta + x_o & (3D \text{ Map to} \\ (2) \quad y &= y_o - y_m & 3D \text{ Global}) \\ (3) \quad z &= x_m \sin \theta + z_m \cos \theta + z_o \end{aligned}$$

The global coordinates  $(x, y, z)$  then produce image coordinates  $(x_I, y_I, z_I)$



a)

PT	COORDS
0	(0,0,0)
1	(3,0,0)
2	(3,0,2)
3	(0,0,2)
4	(1,1,0)
5	(3,1,0)
6	(2,1,2)
7	(0,1,2)
8	(0,1,1)
9	(1,1,1)
10	(1,2,1)
11	(0,2,1)
12	(0,2,0)
13	(1,2,0)

b)

Figure 4-12. Object model defined by a set of vertices and edges.

via the direct perspective transformation.

$$(4) \quad x_I = \frac{fx}{f+y} = \frac{f(x_m \cos \theta - z_m \sin \theta + x_0)}{f + y_0 - y_m} \quad (3D \text{ Global}$$

$$(5) \quad y_I = 0$$

to  
2D Image)

$$(6) \quad z_I = \frac{fz}{f+y} = \frac{f(x_m \sin \theta + z_m \cos \theta + z_0)}{f + y_0 - y_m}$$

Here we assume that a given vector  $(Ax_m, Ay_m, Az_m) - (Bx_m, By_m, Bz_m)$  in the map corresponds to a given vector  $(Ax', Az') - (Bx', Bz')$  in the image. We develop computational formulas for determining map orientation parameters  $(\theta, x_0, y_0)$  from this correspondence.

Rearranging the imaging equations (4) and (6) from above we have the following.

$$(7) \quad \frac{x_I(f - y_m + y_0)}{f} = x_m \cos \theta - z_m \sin \theta + x_0$$

$$(8) \quad \frac{z_I(f - y_m + y_0)}{f} = x_m \sin \theta + z_m \cos \theta + z_0$$

Since (7) must hold for both points A and B, we get two equations from which  $x_0$  can be eliminated, leaving only  $\theta$  unknown.

$$(9) \quad \frac{Ax_I(f - Ay_m + y_0)}{f} - \frac{Bx_I(f - By_m + y_0)}{f} = (Ax_m - Bx_m) \cos \theta + (Bz_m - Az_m) \sin \theta$$

Equation (9) is of the form

$$c = d \cos \theta + e \sin \theta$$

where we make the substitutions  $w = \sin \theta$  and  $\sqrt{1-w^2} = \cos \theta$ .

Thus a standard quadratic equation in  $w$

$$(10) \quad (e^2 + d^2)w^2 - (2ce)w + (c^2 - d^2) = 0$$

is obtained where the coefficients are gotten as follows:

$$c = (Ax_I (f - Ay_m + y_o) - Bx_I (f - By_m + y_o)) /$$

$$d = Ax_m - Bx_m$$

$$e = Bz_m - Az_m.$$

Solving the quadratic yields

$$(11) \quad w = \frac{ce + d\sqrt{e^2 - c^2 + d^2}}{e^2 + d^2}$$

$$(12) \quad \theta = \sin^{-1} w.$$

Knowing  $\theta$ , (7) and (8) can be used to solve for  $x_o$  and  $y_o$  using either A or B point coordinates. For example,

$$x_o = Ax_I(f - Ay_m + y_o) / f - Ax_m \cos \theta + Az_m \sin \theta.$$

Since there is mathematical ambiguity in  $\theta$  from (11) two parameter sets  $(w_1, x_{01}, y_{01})$  and  $(w_2, x_{02}, y_{02})$  can result. It is easy to check for correct parameters using (4) and (6) and the 2 known pairs of corresponding points  $(A_I, A_m)$  and  $(B_I, B_m)$ .

There are two significant cases to note where the computation breaks down. First of all, the discriminant of the quadratic can be negative and hence no  $\theta$  can be obtained. This will happen whenever the map edge cannot possibly be imaged onto the image edge. Few pairings are actually possible due to the fixed scale imposed by  $f$  and  $y_o$ . Secondly, whenever the map edge is vertical both  $d$  and  $e$  above are zero and equations (11) and (12) cannot produce  $\theta$ . Physically we can rotate the object in the map freely about that vertical edge without altering its image and thus we should not expect to get mathematically either.

Figure 4-13 shows a planar section containing the camera axis, the vertical map vector, and hence the image vector as well. Clearly, free rotation about the axis AB will not change this picture. It is also clear, however, that locational parameters  $x$  and  $z$  are completely specified

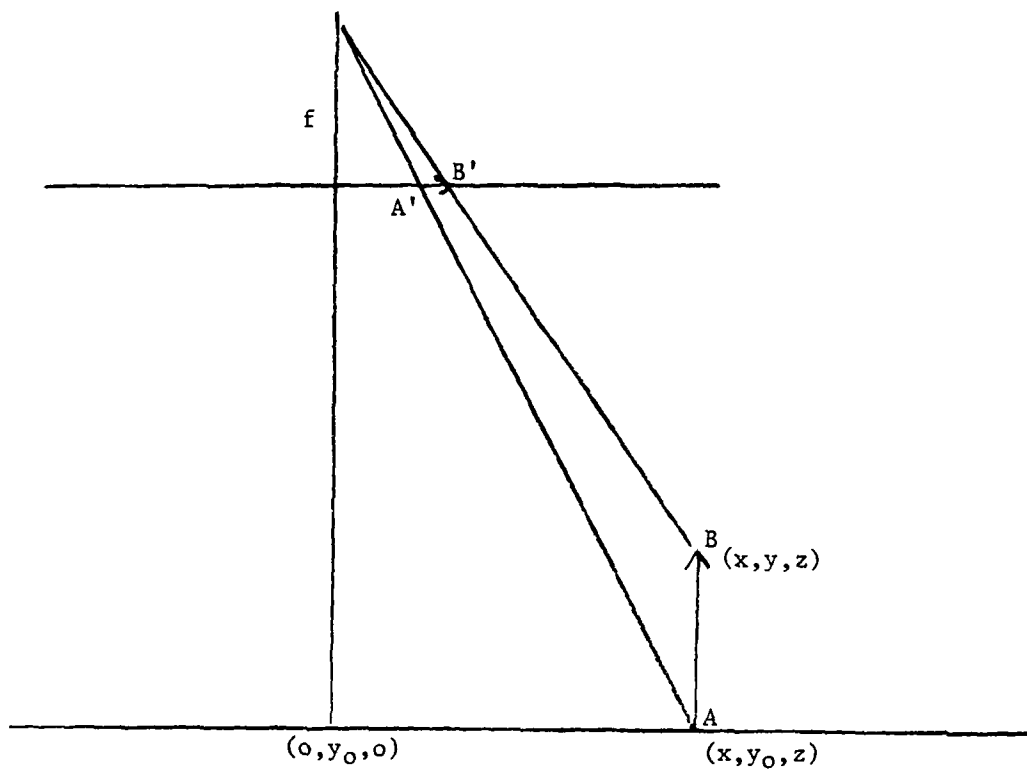


Figure 4-13. Planar section of imaging environment containing both the camera axis and a verticle model vector  $AB$ .

by A'B' and AB provided that they correctly match via the viewing transformation. From the imaging equations (4) and (6) applied to both points A and B we get

$$Ax_I(f+y_o) / f = x = Bx_I(f+y) / f \text{ or}$$

$$(23) \quad Ax_I(f+y_o) = Bx_I(f+y)$$

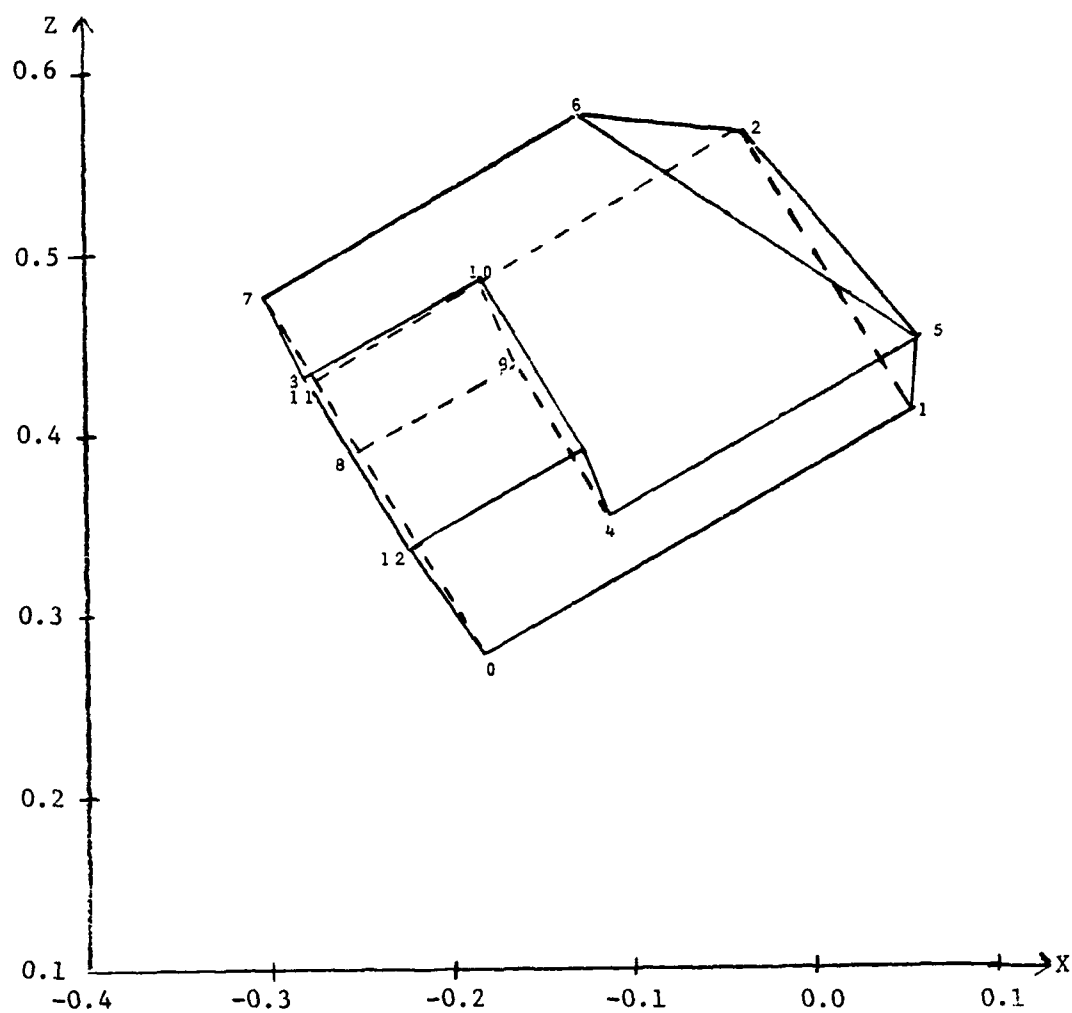
and similarly for the z coordinate relations

$$(24) \quad Az_I(f+y_o) = Bz_I(f+y).$$

Conditions (23) and (24) must hold if A'B' is to possibly match with AB.

We are already in trouble here if only real edge segments are available because (23) and (24) are scaling equations. On the other hand, if abstract vertical map edges which have accurate tips and tails are being used, it would be easy to consider only nonvertical vectors as before. All that need be done is to construct nonvertical vectors by mixing the tip and tail points of several vectors. For this reason, special treatment of vertical vectors can be ignored (justifiably) in the computer programs.

As an example, suppose the map contained the object shown in Figure 4-12. Suppose the image was formed by viewing the object with aspect parameters  $(f=1, y_o=10)$  and orientation parameters  $(\theta=30^\circ, x_o=-2, z_o=3)$ . The resulting image is shown in Figure 4-14, along with the coordinates of the points. Let the object in the map and image be represented by the vectors listed in Table 4-7. Matching the 10x10 pairs of vectors  $(v_i, v_j)$  yields only 12 feasible parameter sets, 10 of which form a cluster about the correct parameters  $\alpha = (\theta=0.525, x_o=-2, z_o=3)$ .



<u>Pt</u>	<u>New Coords: (X,Z)</u>
0	(-.182, .273)
1	(.054, .409)
2	(-.037, .567)
3	(.273, .430)
4	(-.113, .350)
5	(-.060, .450)
6	(-.127, .573)
7	(-.300, .473)
8	(-.250, .387)
9	(.163, .437)
10	(.182, .485)
11	(-.278, .430)
12	(-.222, .333)
13	(-.126, .389)

Figure 4-14. Object model defined by set of vertices and edges.

Table 4-7

Representation of model and image by 10 vectors each, chosen from Figures 4-12 and 4-14.

MODEL VECTOR			IMAGE VECTOR		
Number	Tip	Tail	Number	Tip	Tail
1	(-0.182, .273)	(.054, .409)	1	(0, 0, 0)	(3, 0, 0)
2	(0.054, 0.409)	(-0.037, 0.547)	2	(3, 0, 0)	(3, 0, 2)
3	(-0.037, 0.567)	(-0.273, 0.430)	3	(3, 0, 2)	(0, 0, 2)
4	(-0.183, 0.273)	(-0.273, 0.430)	4	(0, 0, 0)	(0, 0, 2)
5	(-0.113, 0.350)	(0.060, 0.450)	5	(1, 1, 0)	(3, 1, 0)
6	(-0.037, 0.567)	(0.060, 0.450)	6	(3, 0, 2)	(3, 1, 0)
7	(0.060, 0.450)	(-0.127, 0.573)	7	(3, 1, 0)	(2, 1, 2)
8	(-0.127, 0.573)	(-0.300, 0.473)	8	(2, 1, 2)	(0, 1, 2)
9	(-0.037, 0.567)	(-0.127, 0.573)	9	(3, 0, 2)	(2, 1, 2)
10	(-0.300, 0.473)	(-0.250, 0.387)	10	(0, 1, 2)	(0, 1, 1)
					PT 0 TO PT 1
					PT 1 TO PT 2
					PT 2 TO PT 3
					PT 0 TO PT 3
					PT 4 TO PT 5
					PT 2 TO PT 5
					PT 5 TO PT 6
					PT 6 TO PT 7
					PT 2 TO PT 6
					PT 7 TO PT 8



#### 4.2.4 Clustering

All of the versions of LNK's registration technique require clustering to be done in  $\alpha$ -space. The cluster space is filled with points, each one of which represents the matching of one image element to one map element on the basis of local features alone. A cluster of points in this space represents a globally good transformation that matches many image elements to corresponding map elements. A threshold may be set to determine the strength of an acceptable cluster; for instance, we might demand that half of the map elements be matched by image elements. A unique cluster will typically result but there may be no cluster formed in the case where there is poor feature detection or where the image and map really do not match. On the other hand, several strong clusters can result as in the case where symmetry produces several good matching possibilities. LNK has used two different clustering techniques - hierarchical clustering and variable resolution clustering.

In the hierarchical technique, clustering for  $\alpha=(\Delta\theta,\Delta x,\Delta y)$  was done, first on  $\theta$  alone, and then in  $(\Delta x,\Delta y)$ -space given a fixed  $\Delta\theta$ . This technique was useful for human interaction since  $\theta$ -space could be viewed as a histogram and  $(\Delta x,\Delta y)$ -space as a scatter plot.

The  $\theta$ -space was divided into 360 one-degree bins and the rotation  $\Delta\theta_{im}$  that rotated the image edge to the map edge was entered. After smoothing, the top three peaks of the histogram were chosen for the second step. For each of the transformations that had a  $\Delta\theta$  in one of the three peaks, the  $(\Delta x,\Delta y)$  in that transformation was recorded in  $(\Delta x,\Delta y)$ -space. The resulting scatter plot could then be searched for clusters.

Intuitively, clustering can be done by moving a box around the cluster space to see if there is some position at which an acceptable number  $N$  of points are inside the box. If so, the coordinates  $(\theta,s,\Delta x,\Delta y)$  of that position mark a

cluster center and represent a good registration transformation. A computer implementation can be specified when we choose the size of the box and the set of different positions (bins) at which we will try to fill it.

The ultimate spread of a cluster is related to the precision or resolution of the imaging system and the feature detectors. Suppose the Hough detector is used to detect straight edge elements and the precision is  $\pm 1$  degree in direction. An error of 1 degree, or  $\pi/180$  radians, in  $\theta$  could result in an error of  $r\pi/180$  in the  $\Delta x$  or  $\Delta y$  computed from  $\theta$  as in Figure 4-9, where  $r$  is the distance from the origin to the line formed by the straight edge. If  $r$  is 900 pixels then the error induced in  $\Delta x$  and  $\Delta y$  could be 15 pixels. Thus in this case the box (bin) size should never be smaller than 2 degrees x 30 pixels x 30 pixels. In the variable resolution technique binning implementation grids of  $10 \times 10 \times 10$  boxes were used. Each point  $(\theta, \Delta x, \Delta y)$  was distributed to the appropriate bin. Assuming that the ranges of  $\theta, \Delta x$ , and  $\Delta y$  were 360 degrees, 2000 pixels, and 2000 pixels respectively, the bin size of level one would be 36 degrees x 200 pixels x 200 pixels. If any of the 1000 bins at level one received  $N$  or more triples binning was repeated on only those points at level two, and with a bin size of 3.6 degrees x 20 pixels x 20 pixels. A third level of clustering was seldom warranted and in the case at hand would imply a box size of 0.36 degrees x 2 pixels x 2 pixels.

There is one final note on the clustering procedure. Since the cluster space is quantized by the bins, overlapping sets of bins are actually used so that clusters are not lost by being split by bin boundaries. Thus each triple  $(\theta, \Delta x, \Delta y)$  is actually distributed to several bins rather than one.

To use detected straight edges instead of abstract edges, some changes need to be made in the clustering algorithm. Real edge detectors create edges with two general defects. First of all, only small segments of the true edge

are usually detected primarily because only a fixed size detector is used. Second, the detectors may overshoot the true termination of an edge at a corner.

In order to deal with the above problems when real edges are used, the matching of image vectors to map vectors must be made to be sloppy. Figure 4-15 gives the details. The image edge element must be allowed to slide along the potentially matching model edge. Mathematically this generates an infinite set of triples  $(\Theta, \Delta x, \Delta y)$  constrained to be on a line segment in cluster space. In practice the endpoints  $(\Theta, \Delta x_1, \Delta y_1)$  and  $(\Theta, \Delta x_2, \Delta y_2)$  of the line segment, are saved for clustering. Whenever points  $(\Theta, \Delta x_1, \Delta y_1)$  and  $(\Theta, \Delta x_2, \Delta y_2)$  are in separate bins, points are also contributed to all bins in between them. This technique was used for all real edge vectors in the experiments reported in Section 4.2.

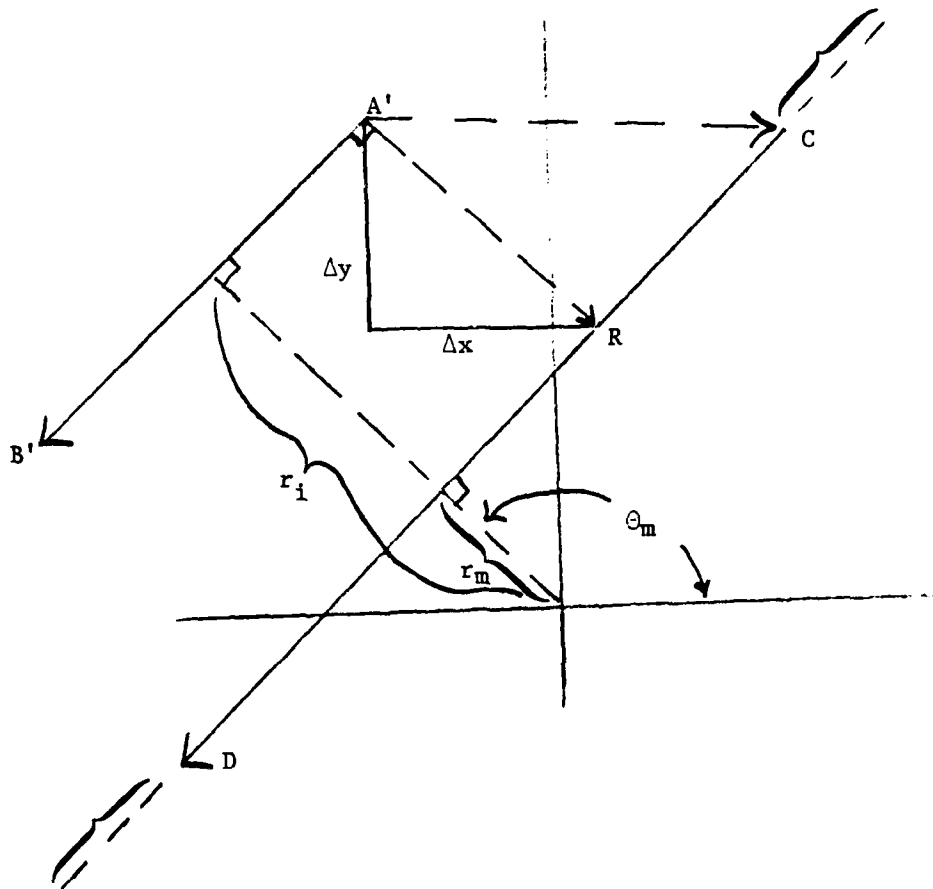


Image edge AB or  $(\theta_i, r_i)$  is rotated  $\theta_m - \theta_i$  onto A'B' or  $(\theta_m, r_i)$  to be parallel to map edge CD or  $(\theta_m, r_m)$ . Projection of A'C onto A'R has constant length yielding equation relating  $\Delta x$  and  $\Delta y$  (translation parameters)

$$\Delta x \cos \theta_m + \Delta y \sin \theta_m + (r_i - r_m) = 0.$$

$(\Delta x, \Delta y)$  are further constrained because point A' should translate no further than point C and point B' should translate no further than point D. Some tolerance can be given on either end since edge detectors can overshoot corners.

Figure 4-15. Constraints on points  $\alpha = (\theta, \Delta x, \Delta y)$  in cluster space derived by pairing real image edge segment AB with real map edge segment CD.

### 4.3 Region Matching

Symbolic matching of images consists of reducing two images to a pair of abstract structures and comparing these structures. The methods we consider here take the regions in a segmented image as components of the structure describing the image. Relationships between regions may be stored explicitly in a regions description, or they may be represented as edges in a graph. Region descriptions may include such features as size, shape and average gray level. All matching procedures based on regions rely on the segmentation procedures used to define the regions. Poor region segmentation can cause serious mismatching of regions. The choice of features describing regions may also be critical in the design of those procedures.

Most region matching algorithms do not provide the registration transformation. This means additional processing must be done, after the region matching procedures are through, to find the desired transformation. Since region segmentation, region features and region matches can all be computationally expensive to obtain, region matching should be approached with caution. The expense might be justified if image interpretation is desired where region descriptions could help tremendously.

The output of a region matching procedure applied to two images is a correspondence between some subset of the set of regions in the first image and a subset of the set of regions in the second image. If we label the regions in image 1 with the symbols  $a_1, \dots, a_N$  and the regions in image 2 with the symbols  $b_1, \dots, b_M$ , then a matching assigns an element of the set  $\{b_1, \dots, b_M, c\}$  to each  $a_i, i=1, \dots, N$ . If  $b_j$  is assigned to  $a_i$  the  $b_j$  corresponds for  $a_i$ . If  $c$  is assigned to  $a_i$  then  $a_i$  doesn't correspond to a region in image 2. We have been unable to find algorithms in the literature for determining a registration from a matching of regions.

Prior to constructing such algorithms, we must have a criterion for the quality of a registration obtained from region matching. One natural criterion is the maximization of the total area matched under a registration. If region  $a_i$  matches with  $b_j$ , then under a registration, we can view both  $a_i$  and  $b_j$  as regions in the same plane so their matched area is  $a_i \cap b_j$ .

If a measure of matched quality based on feature values of  $a_i$  and  $b_j$  is available, then the matched area may be weighted by the match quality. This criterion could be used with a hill climbing optimization technique and an initial registration to provide a search technique for a registration. This procedure could be very costly if the initial registration is poor.

#### 4.3.1 Region Image Matching Using Similarity of Region Features

Region image matching can be done by finding a set of features which describe the regions and then pairing regions which have the best matching set of features. It is desirable that the decisions take into account the adjacency information of the regions, so that adjacent regions in the image match adjacent regions in the map.

One such method of region matching yields a measure of similarity between pairs of regions, one from each of the two images to be matched [Price and Reddy 1979]. This method makes no assumptions about the relative displacement and orientation of the pictures. The steps of their algorithm are as follows:

- 1) Segment the image.
- 2) To each region  $i$  assign a set  $V_{i1}, \dots, V_{in}$  where  $V_{ij}$  denotes the value of the  $j$ th feature for region  $i$ . These numbers may describe features such as shape, size, position, spectral values, etc. To each pair of regions, region  $i$  from image 1 and region  $j$  from image 2, define the region to region match rating,  $R_{ij}$ , by:

$$R_{ij} = - \sum_{k=1}^n \left| V_{ik} - V_{jk} \right| W_k S_k$$

where  $W_k$  is a normalization factor for the  $k^{\text{th}}$  feature and  $S_k$  is a measure of importance of the  $k^{\text{th}}$  feature. Larger values of  $R_{ij}$  indicate good matches.

- 3) In this step we attempt to improve the accuracy of the rating  $R_{ij}$  by taking into account adjacency information. To each region  $R_i$  in image 1, assign the region  $R_j$  in image 2 which maximizes  $R_{ij}$ . Let  $N_1$  denote the set of regions in image 1 which are neighbors of region  $R_1$  and have a match in image 2. We say two regions are neighbors if

they have a common boundary point. Let  $N_j$  denote those regions in image 2 which match the regions in  $N_i$ . Let  $N_j'$  denote those regions in  $N_j$  which are neighbors of  $R_j$ . The neighbor feature value (NFV) of  $R_i$  is defined to be the number of elements in  $N_i$  and the NFV for  $R_j$  is the number of elements in  $N_j'$ . Recompute the  $R_{ij}$ 's using this additional feature and assign to each region in image 1 the region in image 2 which matches it best.

This matching algorithm is designed to be invariant under rotation and translation of the images. By omission of size features, the algorithm can be made invariant to scale change. Dissimilar image matching and image to map matching can be handled by this method.



#### 4.3.2 Region Adjacency Graph

One method of symbolic matching is to represent region segmented images or maps as graphs and to use graph theoretic techniques to do the matching [Pavlidis 1977]. A graph is a set  $\{V_i\}$  whose elements are called nodes and a collection of pairs from  $\{V_i\}$  called edges. For scene matching, each node corresponds to a region and an edge in the graph corresponds to two regions which are adjacent in the image. The graph thus formed is called the region adjacency graph (RAG) of the image. Each node contains a set of labels describing the region represented. These labels give information about shape, size, average brightness, etc. The RAG of a map database may contain additional information in the node labels such as the type of region, e.g. forest, fields.

Denote the nodes of the RAG for image 1 by  $\{N_i\}$  and the nodes of the RAG for image 2 by  $\{M_i\}$ . We ultimately wish to match elements of  $\{N_i\}$  with elements of  $\{M_i\}$ . Thus we must define a notion of matching between the labels attached to nodes  $N_i$  and  $M_j$  for any  $i$  and  $j$ . The notion of label matching is problem dependent, but typical matching conditions for two regions would be to have approximately the same area, shape and average brightness. Poor segmentation may necessitate a more involved definition of matching. We would also like to require the condition: if node  $N_i$  matches  $M_j$  and  $N_k$  matches  $M_p$  and  $N_i$  and  $N_k$  are adjacent, then  $M_j$  and  $M_p$  should be adjacent.

The problem of finding the largest number of matching nodes is called the subgraph isomorphism problem and is, in general, computationally expensive. However good segmentation and well-chosen labels could reduce the cost considerably. In addition, the cost could be reduced by ignoring adjacency information or by finding a, not necessarily good, approximate registration first.

AD-A102 619

L N K CORP SILVER SPRING MD  
STUDY OF DIGITAL MATCHING OF DISSIMILAR IMAGES. (U)  
NOV 80 B A LAMBIRD, D LAVINE, G C STOCKMAN

F/6 9/2

DAAK70-79-C-0234

UNCLASSIFIED

ETL-0248

NL

2 OF 2

ADA  
10/2/81

END  
DATE  
FILMED  
9-81  
DTIC

In the latter case, a prerequisite for label matching is that regions should be in approximately the same location under the registration.

In either case, an approximate solution can be obtained using a maximal network flow algorithm [Frank and Frishch, 1971]. Under either of the above assumptions, the RAG's for the two images have no edges. To apply the maximal flow algorithm, we form a new graph  $G$  with vertices  $\{M_i\} \cup \{N_j\} \cup \{\text{source}\} \cup \{\text{sink}\}$  where source and sink are two new nodes. There is an edge from source to each  $M_i$ , and an edge from each  $N_j$  to the sink. For each  $i$  and  $j$  there is an edge from  $M_i$  to  $N_j$  if node  $M_i$  can be matched with  $N_j$ . In addition, we assign the integer one, called the capacity or weight, to each edge of the graph. Finally an optimal flow algorithm is applied to the graph. The output of this program is an assignment of either zero or one to each edge of the graph. If we erase the source and sink and all edges which have been assigned a value zero, then the remaining graph will have the property that no node has more than one edge incident to it. If we erase nodes having no edges, then the remaining edges define a one-to-one correspondence between a subset of the regions of one image with a subset of the regions of the other image. This correspondence has the property that it maximizes the number of regions matched.

The above method assumed that each region in an image could correspond to at most one region in the map or other image. Under real-world conditions, this assumption is unlikely to hold as it would be quite possible, as an example, for a forest which is one region in a map, to be segmented into several regions in the image. The above algorithm could handle this possibility by changing the initial weights of the edges, connecting the sink with nodes  $N_i$ , to a number greater than one. If the  $\{M_i\}$  are the map regions and  $\{N_i\}$  are the image regions, this will allow a many-to-one relation from the map to the image.

The RAG is designed to be invariant under rotation and translation of the images. If features used are based on shape and not size, then the RAG is also invariant under scale change. The RAG can be used for matching an image to a map or for matching images from dissimilar sensors.

#### 4.4 Comparison of Registration Procedures

Not all of the methods could easily handle the full RS&T transformations. While the LNK registration procedure can easily handle the full RST transformation, most of the other algorithms could not. Of the correlation-based methods, only the invariant moment correlation is invariant under RST transformations. It is possible to extend the other correlation-based methods to include rotation by repeating the correlation while rotating the image with respect to the map. Except for the hierarchical technique, where the rotation could be done at low resolutions, this extension would be extremely time consuming. The region matching algorithms, with careful selection of region features, are also invariant under RST transformations.

The space requirements for storing the map varies greatly, even within one registration technique, depending on the features being used. With correlation, if a whole gray-scale image is used as the map, then there is no data compression. Whereas if edges are used for correlation-based techniques or LNK's registration procedure, then only the chain codes (if the edges are not straight) or the endpoints (if the edges are straight) need be stored. This could lead to a fair amount of data compression. If only point features are used with correlation or LNK's procedure, then a fair amount of data compression is again possible.

The space requirements for a map using a region matching procedure are highly variable. It largely depends on the number and type of features used to characterize the regions. What is needed to be stored is the adjacency relationships and the features characterizing each region. As long as none of the features are space consuming, such as the chain code of the boundary, then the data compression could be large.

In all of the techniques, the amount of data compression must be balanced against the need for accuracy. It is better to allow the map to have a large number of features and let the feature extraction on the image be less extensive but reliable. This means that any image feature, which should be reliable, would have a corresponding map feature but not necessarily vice versa.

The relative speeds of the registration techniques are highly variable. It is expected that LNK's registration procedure would perform the fastest; straight correlation would be very slow. Sequential and hierarchical correlation, while three orders of magnitude faster [Hall 1979] than straight correlation, would be slower than LNK's procedure. Since region matching methods require image segmentation, they are potentially very slow. In addition, since more processing must be done to find the registration transformation, region matching becomes a very slow process.

The robustness of the registration techniques is an important feature. In all cases, the robustness of any registration technique is going to depend on the robustness of its corresponding feature detector. It is difficult to say much about the robustness of the procedures without experimentation, but a few things are known. It is known that correlation techniques using gray scale images are highly susceptible to noise. Correlation techniques using edges which are not fattened also have difficulties.

LNK has performed enough experimentation with the LNK registration procedure to know that with mediocre feature detection, the registration technique performed well. The region matching procedures are very heavily dependent upon the region segmentation. Poor region segmentation would make region matching almost impossible for most cases.

It might be justifiable to use a more time consuming or more space con-

suming registration procedure if the extra time or space produces extra information which can later be used for scene analysis. For example, while finding and classifying intersections is more time consuming than finding just edges, it can help in later identification of roads and buildings.

#### 4.5 Image Disparity Determination

Once a global registration has been determined, it usually is necessary to modify the transformation to account for local distortion. The distortion could be caused by motion, temporal scene changes, imperfect image rectification, or local altitude variations. The disparity between two registered images is the small local differences in corresponding pixel locations caused by these distortions. The analysis of image disparity in similar scenes is a key element in the computer analysis of motion as well as in the construction of depth information from stereograms.

Barnard and Thompson [1980] give an algorithm which takes as input, two images and an approximate global registration, and outputs a set of point features from each image and a 1-1 map identifying the correspondence between these two sets. This correspondence can be used to define a piecewise linear transformation between the images which refines the global registration. Three properties of images play a critical role in determining the success of their approach.

- (1) The measure of distinctness of points.
- (2) The measure of similarity of such points taken from the two images.
- (3) The measure of how well a point match accords with matches of neighboring points.

Point features lie in highly variable areas in which the variation is large in all directions from the point. An interest operator given by Moravec [1977] defines a point to be of high interest, if there is a large variance of the gray levels along the horizontal, vertical and diagonal directions within the 5x5 neighborhood about the point. The interest operator tentatively assigns the minimum of these variances as the interest value of the point. Finally local maxima with interest values above some threshold are selected as the



true points of interest.

The approximate global registration is applied to the two images. A relaxation procedure is then applied to match points in the two images. Each interest point,  $a$ , in image 1 is assigned a set of labels and a corresponding set of disparity vectors. The labels are the points  $\{b_i\}$  in image 2 lying in a small neighborhood about  $a$ 's transformed location. The disparity vector corresponding to label  $b_i$  is the vector from point  $a$  to point  $b_i$ . The merit of a label  $b_i$  is based on the similarity of the  $5 \times 5$  windows surrounding  $a$  and  $b_i$ . An updating rule is then applied to the merits as follows. The merit of a label  $b_i$  is increased when nearby points in image 1 have labels of high merit with disparity vector similar to that of  $b_i$ . Barnard and Thompson considered ten iterations of this procedure sufficient.

This procedure is not restricted to points derived using Moravec's interest operator. Any of the point features described in section 3.3 could be used. In particular, the Moravec interest operator is unlikely to work well when registering images from different sensors, since it is dependent upon gray scale. However, problems involving registering images to maps may be solved using this method.

In the case of matching images from different sensors, the notion of similar neighborhoods for corresponding points would have to be redefined since gray levels may not match well. Similarity of neighborhood texture might provide a feasible alternative to grey level matching. In all cases, if the feature points are too sparse in the images, the procedure for updating the merits will require considerable revision since small neighborhoods about the feature points may not contain other feature points.

## 5. Conclusions and Recommendations

The preceding sections report on L.N.K.'s study of the problem of registration of images with maps or dissimilar images. The registration process is divided into four steps:

1. Determining and extracting the appropriate features.
2. Computing the parameters of an approximate global registration transformation.
3. Finding the disparity for a subset of points in the images.
4. Determining a global nonlinear transformation for the entire image.

Three classes of features and their detectors are presented in this report: edge features, point features, and region features. Various methods for registration are described in the preceding sections including several correlation procedures, the L.N.K. registration technique and its extension, and some region matching procedures. The registration techniques are briefly compared and contrasted. Their ability to handle the full RS&T transformations and their time and space requirements are examined.

## 5.1 Conclusions

Based on L.N.K.'s experience with feature detection and registration and the reported success of various algorithms presented in the literature, the conclusions of this study are the following:

- . Edge features or edge-based point features, such as intersections and high curvature points, are the most promising features for registration.
- . Abstract edges and triangles formed from point features are very useful for registration. They can provide a more accurate initial transformation for little additional feature detection effort. In particular the clustering in L.N.K.'s registration procedure will be performed faster using abstract edges than real edges.
- . Only two of the registration techniques studied are worthy candidates for further investigation. L.N.K.'s registration procedure can provide a full RS&T transformation and still work well even with only fair feature detection. The other promising technique is a combined hierarchical-sequential correlation method using edge images.
- . Map-guided registration can aid in the registration process by providing as part of its content what feature to use, the particular feature detector and appropriate window size and threshold information, and which registration method to use. (For example, if the map contains a scene with man-made structures then intersections would be a good feature, whereas in scenes with few man-made structures high curvature points could be used.) The use of maps in guiding the decision making enables the process to be tailored to the given situation.
- . The determination of the disparity for a subset of points in the image

is an important open problem. (One method is described in this report but much more work on this subject needs to be done.)

## 5.2 Recommendations

Based on this study the following theoretical and experimental investigations are proposed.

- I. L.N.K. recommends that the following feature detection algorithms be implemented on the DIAL system and tested:
  - a. Hough edge detector
  - b. Laplacian (of the Gaussian function) edge detector
  - c. Linking of edges algorithm
  - d. Intersection detection algorithm
  - e. High curvature points algorithm
  - f. Moravec interest operator algorithm
- II. The accuracy and robustness of a registration procedure are dependent upon the corresponding feature detectors. Various facets of this dependency need to be investigated. Two of the factors are the density and distribution of the features in the map and image. L.N.K. recommends that some experiments be done to estimate the minimum requirements on the number and distribution of features.

This experimentation would proceed in two ways. One approach first involves human digitization of the desired features, such as high curvature points or intersections from an image. A map of the image would then be constructed by digitizing features from a different image, but of the same scene, whose registration with the first image is known. The accuracy of the registration process would then be examined by studying the effect of varying the distribution and number of features. The experiment would be repeated with several sets of images so that more valid conclusions may be reached.

The second approach is more elaborate. For an imaginary scene a

terrain model would be constructed wherein every point would be represented by its three coordinates  $(x,y,z)$ . An image of the scene would be constructed from the terrain model by performing a perspective and projective transformation on the map points. An arbitrary RS&T transformation on the image would then be calculated. Feature points on the map and image would be selected and the result of varying their distribution and density on the registration process would be investigated.

III. L.N.K. recommends that in parallel with the above experiments, a mathematical analysis of the registration process be conducted to investigate the sensitivity of the L.N.K. registration procedure to "noise". A simplified approximation to the problem is to consider two identical isolated point images (such as high curvature point images) and determine the maximal variation in the ideal rotational and translational parameters as a function of the size of a perturbation of points in one of the images. This variation will also depend on the length of the abstract vectors constructed by the L.N.K. procedure. This analysis would be used in designing procedures for selecting abstract vectors for the L.N.K. registration procedure. It would also be used to determine the expected "noise" of the procedure, and the accuracy of the procedure as a function of its noise.

IV. The LNK registration procedure requires clustering to be performed in the transformation parameter space. LNK has only performed clustering on three parameters. The clustering algorithm needs to be extended to four parameters to handle the full RST transformations. LNK recommends that further investigation and error analysis of clustering algorithms be done and that the effect of varying the size and dispersion of the cluster on the algorithms be investigated experimentally using existing data.

- V. L.N.K. recommends that a set of hierarchical edge image registration algorithms be devised and tested. The algorithms would find successively smaller windows with lower resolution from an original edge image. Registration would be performed by matching the smallest windows, which have the lowest resolution, with the full-sized map, which also has low resolution. After finding tentative matches at low resolution larger windows at high resolution would be examined to select a good registration.

One method of matching the window with the map should use the LNK registration procedure. A second method should use the sequential correlation technique. The hierarchical approach may speed up the registration process.

- VI. LNK recommends that interleaving of registration and feature extraction be investigated. All registration procedures presented in this report assume that all feature extraction has been done prior to the time of registration. To reduce the cost of feature extraction and registration it may be desirable to determine an approximate registration from a limited set of features and then use the approximate registration to locate areas suitable for further feature extraction. These features can then be used to obtain an improved registration. This procedure can be iterated until a single good registration arises.

In the case of image to map registration, the map can be used to specify areas in the image which are likely to provide good edges under the approximate registration. For image to image registration one can do extensive edge detection in one image and use this to guide the search for good edges in the other image. The LNK procedure could be modified to this interleaved form.

Several important questions must be investigated before the above interleaved procedure can be effectively applied. Methods need to be

devised and investigated for selecting portions of a map having features which are likely to prove useful for quick registration. This selection process will involve defining various edge and point measures such as the number of high curvature points in the neighborhood of a given high curvature point. A more elaborate measure might involve simple statistics on the distribution of abstract vector lengths and orientation in the neighborhood of a point. These and other measures could be used in the selection of the type of feature extractor and the feature extractor parameters to be applied in a given portion of the image.



## References

- D.I. Barnea and H.E. Silverman  
A Class of Algorithms for Fast Digital Image Registration  
IEEE TC Vol C-21 Feb. 1972.
- M.A. Crombie  
Semiautomatic Pass Point Determination Using Digital Techniques  
U.S. Army Engineers Topographic Laboratories Report  
ETL-0051, 1975.
- L. Davis  
A Survey of Edge Detection Techniques  
Computer Graphics and Image Processing, Vol. 4, 1975, pp. 248-270.
- R. Duda and P. Hart  
Use of the Hough Transform to Detect Lines and Edges in Pictures  
CACM Vol. 15 No.1 Jan. 1972, pp. 11-15.
- R. Duda and P. Hart  
Pattern Classification and Scene Analysis  
Wiley-Interscience, New York, 1973.
- S.A. Dudani, A.L. Luk, P.P. Stapudd, C.S. Clark and B.L. Bullock,  
Model-Based Scene Matching  
Hughes Research Report #509, July 1977.
- H. Frank and I. Frisch  
Communication, Transmission, and Transportation Networks  
Addison-Wesley, Reading, Mass., 1971.
- H. Freeman  
Computer Processing of Line-Drawing Images  
Computing Surveys Vol. 6, 1974, pp. 57-97.
- A. Gilchrist  
The Perception of Surface Blacks and Whites  
Scientific American 240, 1979, pp. 88-97.
- E. L. Hall  
Computer Image Processing and Recognition  
Academic Press, New York, 1979.
- K. Hayes and A. Rosenfeld  
Efficient Edge Detectors and Applications  
Computer Science Center TR-207  
University of Maryland, 1972.
- B. Horne and B. Bachmann  
Using Synthetic Images to Register Real Images with Surface Models  
M.I.T. A.I. Memo 437, August, 1977.

- T. Kanade  
Region Segmentation: Signal vs. Semantic  
Computer Graphics and Image Processing Vol. 13, 1980, pp. 279-298.
- D. Marr and E. Hildreth  
Theory of Edge Detection  
A.I.M 518, M.I.T., April, 1979.
- D. Milgram  
Edge Point Linking Using Convergent Evidence  
Proceedings of the Image Understanding Workshop, Arpu, November, 1978.
- H. Moravec  
Towards Automatic Visual Obstacle Avoidance  
Proceedings of the 5th Joint Conference on Artificial Intelligence,  
August, 1972, p. 584.
- T. Pavlidis  
Structural Pattern Recognition  
Springer Verlag, New York, 1977.
- K. Price  
Change Analysis and Scene Detection in Multispectral Images  
Ph.D. Thesis Carnegie-Mellon University, Pittsburgh, PA, Dec 1976.
- H. K. Ramapriyan  
A Multilevel Approach to Sequential Detection of Pictorial Features  
IEEE TC Vol C-25 Jan. 1976.
- E.M. Riseman and M.A. Arbib  
Computational Techniques in the Visual Segmentation of Static Scenes  
Computer Graphics and Image Processing Vol. 6, 1977, pp. 221-276.
- A. Rosenfeld and M. Thurston  
Edge and Curve Detection for Visual Scene Analysis  
Computer Science Center, TR-207  
University of Maryland, November, 1972.
- A. Savole, A. Wilsmeier, E. Noges, J. Geros  
Development of an On-board Navigational Update System Using Pattern  
Recognition  
IEEE Pattern Recognition Conference, Chicago, May, 1978.
- G. Stockman and A. Agrawala  
Equivalence of Hough Curve Detection and Template Matching  
CACM Vol. 20, No.11, November, 1977.
- G. C. Stockman and S. H. Kopstein  
The Use of Models in Image Analysis  
Report #AMRL-TR-78-117. Final Rept. by L.N.K. Corp under  
Contract F33615-76-C-0521 to Aerospace Med. Res. Lab.,  
Wright-Patterson A.F.B., Ohio 45433, Jan 1979.
- R. Y. Wong and E. L. Hall  
Scene Matching with Invariant Moments  
Computer Graphics and Image Processing, Vol. 8, Aug., 1978.

C. T. Zahn  
An Algorithm for Noisy Template Matching  
Proc. IFIP Congress, pp. 698-701.

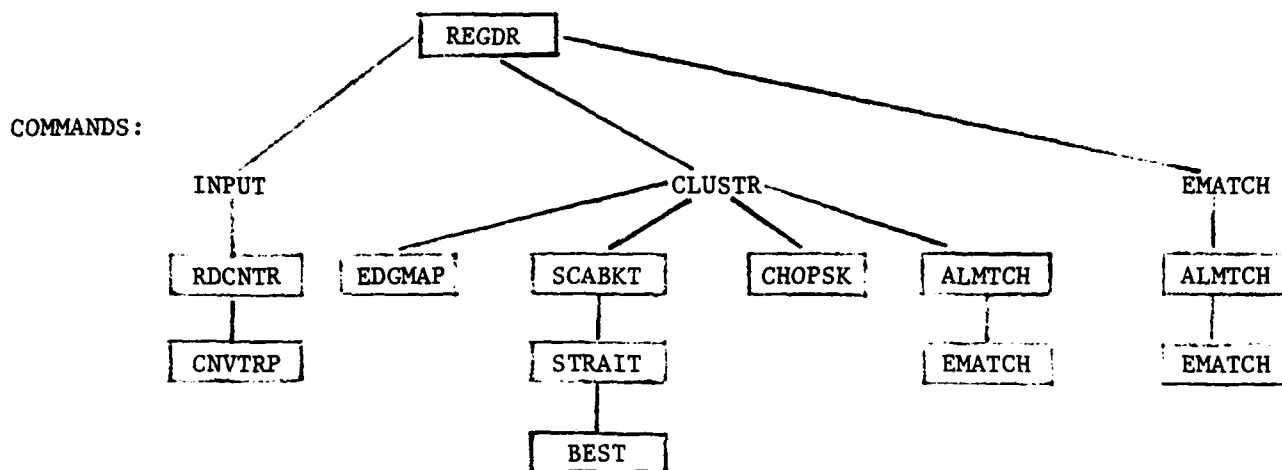
S. Zucker  
Region Growing: Childhood and Adolescence  
Computer Graphics and Image Processing, Vol. 5, 1976, pp. 382-399.

S. Zucker, R. Hummel, A. Rosenfeld  
An Application of Relaxation Labeling to Line and Curve Enhancement  
IEEE TC C-26, April, 1977.

S. Zucker and R. Hummel  
Toward a Low-Level Description of Dot Clusters: Labeling, Edge, Interior  
and Noise Points  
Computer Graphics and Image Processing, Vol. 9, No.3, 1979, pp. 213-234.

## Appendix A LNK Registration Procedure Software

The LNK registration procedure software, as existing on a HP2100MX mini-computer at ETL, consists of one load module containing one main program and several subroutines. The program structure is illustrated below:



The registration program takes a set of image edges from the file **IMAGE** and a set of map edges from the file **MAP** and computes the possible transformations that will transform the image onto the map. The technique used is to cluster the possible transformations in 3-dimensional space and select those with the strongest support.

The data structures used reside in the four common blocks, presented in A.1, and the match weight matrix, **MATCHS(200,30)** is in the common block **MATRIX** which resides in the RTE-IV, extended memory facility.

The command to initiate the 3-dimensional clustering that does the bulk of the registration work, starts in column one and is CLUSTER.

A command that can be presented independently of the CLUSTER command, is the EMATCH command. This command allows the user to present a possible transformation and have the software evaluate the strength of the transformation given the edge information input as a result of the INPUT command processing. This command can be given before, after or in place of the CLUSTER command. The format is:

EMATCH <THETA>, <XS>, <YS>, <DTOL>

where

THETA - rotational angle, in degrees, of the transformation to be evaluated. (integer)

XS - X-shift of transformation to be evaluated (real)

YS - Y-shift of transformation to be evaluated (real)

DTOL - distance tolerance to be used in computing strength of matches for this transformation. (integer)

. The command to terminate processing is simply FINISH.

# D.1 Procs containing common blocks used in registration

```

C
C REGIST PROC*****H.P. REGISTRATION CODE 9FEB79*****
C
      INTEGER X1(200),X2(200),Y1(200),Y2(200),U1(30),
      *      U2(30),V1(30),V2(30),NIMAGE,NMAP,IXCENT,IYCENT,MYCENT,
      *      MYCENT,RSIZE,CSIZE
      REAL THET(2,200),RAD(2,200)

C
C
C X1,X2,Y1,Y2 - ARRAYS FOR ENDPOINTS OF IMAGE LINES
C U1,U2,V1,V2 - ARRAYS FOR ENDPOINTS OF MAP LINES
C THET - ANGLES FOR IMAGE AND MAP
C RAD - RAD FOR IMAGE AND MAP
C NIMAGE - # OF LINES IN IMAGE (# OF ROWS IN MATRIX)
C NMAP - # OF LINES IN MAP (# OF COLUMNS IN MATRIX)
C IXCENT,IYCENT - X AND Y OF IMAGE CENTER
C MYCENT,MYCENT - X AND Y OF MAP CENTER
C RSIZE - # OF ROWS IN MATCHS
C CSIZE - # OF COLUMNS IN MATCHS
C
C NAMED COMMON REGIS IS USED TO HOLD INFO
C
      COMMON /REGIS/ X1,X2,Y1,Y2,U1,U2,V1,V2,THET,RAD,
      *      NIMAGE,NMAP,IXCENT,IYCENT,MYCENT,MYCENT,RSIZE,CSIZE

C
C*** END OF MACRO REGIST, H.P. VERSION 9FEB79*****
C
C
C*****PROC BUCKET H.P. VERSION 9FEB79*****
C
C DEFINES CLUSTER MATRICES. CLUSTERING IN 3-D (THETA,X,Y)
C BKCENT - CONTAINS COUNT OF # OF HITS IN EACH BUCKET
C BKTOFF - CONTAINS COUNT OF # OF HITS IN OFFSET MATRIX
C
      INTEGER BKCENT(10,10,10),BKTOFF(10,10,10)
      INTEGER NZEROT(10),NUMTHE,NUMX,NUMY,NZEROD(10)
      COMMON /CLSTR/BKCENT,BKTOFF,NZEROT,NZEROD,NUMTHE,NUMX,NUMY

C
C*** END OF PROC BUCKET ***
C

```

```

C
C*****PRTELG MACRO H.P. VERSION 9FEB79*****
C
C      INTEGER PRTEINP,PRTEMM,PRTEBUK,PRTESEK
C
C      PRTEINP - PRINT INPUT FLAG (RECTANGULAR AND POLAR)
C      PRTEMM - PRINT MATCH WEIGHT MATRIX FLAG
C      PRTEBUK - PRINT 3-D CLUSTERING MATRIX FLAG (BUCKETS)
C      PRTESEK - PRINT SMOOTHED CLUSTERING MATRIX
C
C      NAMED COMMON PRTEL IS USED TO HOLD FLAGS
C
C      COMMON /PRTEL/ PRTEINP,PRTEMM,PRTEBUK,PRTESEK
C
C*** END OF PROC PRTELG
C
C
C*****PROC CHSDEF H.P. VERSION 9FEB79*****
C
C      INTEGER DX(8),DY(8)
C
C      DX,DY - CHANGES IN X AND Y FOR CHAIN CODES
C      THE NAMED COMMON DELTS HOLDS THESE VALUES AND IS INITIALIZED IN THE
C      BLOCK DATA PROGRAM
C
C      COMMON /DELTS/ DX,DY
C
C*** END OF PROC CHSDEF
C

```

## D.2 Possible commands

The command needed to get things started is the INPUT command.

The format is as follows (starting in column 1):

```
INPUT  <ANGTOL>,<STOL>,<DTOL>,<NUMLEV>,<PRTBUK>,<PRTOFF>,<PRTMWM>,  
      <PWIDTH>,<LENCHK>,<THRESH>
```

Where the input fields have the following significance:

- ANGTOL - angular tolerance (in degrees) between model edge and a rotated image edge. Used in the routine ALMTCH.
- STOL - segment tolerance for generating the line segments in EDGMAP.
- DTOL - distance tolerance used in computing strength of match between a model edge and a transformed image edge. Calculation using DTOL is in routine EMATCH.
- NUMLEV - number of clustering levels desired. Maximum currently possible is 5. Used as a controlling parameter in REGDR.
- PRTBUK - flag for printing original clustering matrix. If value is 1, matrix is printed at each level.
- PRTOFF - flag for printing offset clustering matrix. If value is 1, matrix is printed at each level.
- PRTMWM - flag for printing match weight matrix.
- PWIDTH - number of columns available on output line. Can be either 72 or 132.
- LENCHK - if set to 1, only edges whose lengths are approximately the same will be compared.
- THRESH - minimum number of line segments that must pass through a bucket, at the highest clustering level, in order for that bucket to be considered as a peak.

All above inputs are integer.



Routine Name	Arguments	Function
REGOR	Main Program	Reads and analyzes commands and calls appropriate subroutine. Sets up initial values for variables contained in common blocks found in the procs: REGIST, BUCKET, and PRTFLG.
RDCNTR	(NUMLEV, SCALEX, SCALEY, SCALET, LOXBND, LOYBND, LOTBND)	Reads in the edge information for the image and then the map from DISK FILES. Image edge information must reside in the file IMAGE::18. Map edge information must reside in MAP::18. Computes centers of image and map windows and places them in common block. Using NUMLEV, computes the three scales at each level and stores them in arrays SCALEX, SCALEY and SCALET. Computes lower bounds at first level and stores them in LOXBND(1), LOYBND(1) and LOTBND(1). Image edge information stored in arrays X1,X2,Y1,Y2 of common block REGIS. Map edge information into arrays U1,U2,V1,V2 of same common block.
CNVTTP	(X1,Y1,X2,Y2,RADIUS, THETA)	Takes a directed straight line segment beginning at (X1,Y1) and ending at (X2,Y2) and converts it to polar coordinates (RADIUS,THETA).
EDGMAP	(X1,Y1,X2,Y2,U1,V1, U2,V2,STOL,THETAR, STAILX,STAILY,SHEADX, SHEADY)	Given an image edge defined by (X1,Y1) and (X2,Y2), a map edge defined by (U1,V1) and (U2,V2), an angle of rotation of image to map edge (THETAR) and a tolerance (STOL); a line segment in x-space which represents the constraints of the x-shift and the y-shift in transformations on these two edges is calculated and is represented by (STAILX, STAILY) and (SHEADX,SHEADY).
CLABET	(TAILX,TALLY,HEADX, HEADY,THETA1,THEUND, SCALEX,LOX,SCALEY,LOY, MAINUM,OUTUM)	Creates the clustering matrices used to find the most relevant transformations. Since two sets of clustering matrices are used by the approach (regular and offset matrices) the routine is called twice for each line segment it processes. The line segment is defined by (TAILX, TALLY), (HEADX,HEADY) and the angle of

Routine Name	Arguments	Function
		called to actually evaluate how good each match is. Taking the results of EMATCH, ALMTCH creates the match weight matrix matches (in common block MATRX), the average match weight, MTCHWT, the number of matching rows, NMCHRW, the number of matching columns, NMCHCL, and in which columns a match has been found. If LENCHK=1, only edges of approximately the same length are compared. PWIDTH is used by available output routine.
EMATCH	(SEG1,SEG2,GRAD,THETA, XSHIFT,YSHIFT,DTOL)	Given the transformation (THETA,XSHIFT, YSHIFT), the edge in the image (stored in array SEG1), the edge in the map (stored in array SEG2), the gradient direction of the map edge, GRAD, the gradient direction of the image edge, THETA, and the tolerance, DTOL; computes strength of match and returns a value between 0 and 1.
Worker routines not shown in program structure.		
STACK	(CODE,PEAKS,I)	Maintains a push down stack that keeps track of peaks to be evaluated. Each entry contains 5 fields: THETA INDEX, X-INDEX, Y-INDEX, WEIGHT and which matrix peak appeared in.
PRINT	(PWIDTH,CNTSUM)	Prints either the original or offset clustering matrix depending on the value of CNTSUM. Matrices in common block BUCKET
PRMAT	(WRITE,SIZE,NIMAGE, NMAP,PWIDTH)	Prints the match weight matrix created in ALMTCH.
RDWRIT	(FILLND,OPCODE, BUFFER,IL,LENG, TYPE)	Controls the input and output to the intermediate disk files. Possible operations are OPEN, READ,WRITE,REWIND, and CLOSE. Can have up to 2 files open at once.
SGN	(I)	Returns 1 if I > 0, -1 if I < 0 and 0 if I=0.

Routine Name	Arguments	Function
		the line is in THETA. MATNUM indicates whether incrementation is to be done in the regular matrix (MATNUM=1) or the offset matrix (MATNUM=2). The scale of the matrix is defined by SCALX, LOX, SCALY and LOY. The line is chased from tail to head and each bucket of the matrix that it passes through is incremented. Information about each line is written to an intermediate output file. Clustering matrices BKT CNT & BKTOFF are in the common block in proc BUCKET.
STRAIT	(AX,AY,BX,BY, NNEICB,STORAG,N,NLINKS, IFLAG)	Given a line defined by (AX,AY) and (BX,BY) and whether it is to use 8-directional or 4-direction Freeman codes, STRAIT computes the links along that straight line. The links are stored in the array STORAG(N). If the number of links, NLINK, is greater than N, IFLAG is set to indicate an error.
BEST	(FX,FY,TX,TY,NNEICB L1,L2,N1,N2)	Given a line from (FX,FY) to (TX,TY); computes the straightest path, returning # of L1 links in N1 and # of L2 links in N2.
CHOSPK	(PEAKS, MAXPKS, NUMPKS, PRIFLG,PWIDTH,THRESH)	Scans the two clustering matrices, BKT CNT and BKTOFF, to find the MAXPKS highest values. The indices of the high valued buckets are stored in the array PEAKS with the number of peaks actually chosen being in NUMPKS. THRESH defines the minimum value to be considered as a possible peak. PRIFLG and PWIDTH provide information to the print routines.
EMATCH	(THETA,XSHIFT,YSHIFT, ANGTOL,DTOL,PWIDTH, LENCHK,MATCH, NMCHES,NMCHCL,COLECH)	Given a transformation defined by (THETA,XSHIFT,YSHIFT), an angle tolerance of ANGTOL and a distance tolerance of DTOL; an evaluation of how good the transformation is made. This is done by transforming any image edge that possess approximately the correct gradient onto a map edge. The function EMATCH is

DATE  
FILMED  
-8